AFOSR 64-1155
MIT Fluid Dynamics Research
Laboratory Report No. 64-4

THE CALCULATION OF TWO-AND THREE-DIMENSIONAL INVISCID UNSTEADY FLOWS BY THE METHOD OF CHARACTERISTICS

HARRY SAUERWEIN

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
FLUID DYNAMICS RESEARCH LABORATORY

GRANT NO AFOSR 156-64
FINAL SCIENTIFIC REPORT
JUNE 1964

PREPARED
FOR
OFFICE OF SCIENTIFIC RESEARCH
UNITED STATES AIR FORCE
GRANT NO. AF AFOSR 156-64

AFOSR 64-1055

MIT Fluid Dynamics Research

Laboratory Report No. 64-4

THE CALCULATION OF TWO- AND THREE-DIMENSIONAL INVISCID UNSTEADY FLOWS BY THE METHOD OF CHARACTERISTICS

Harry Sauerwein

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

FLUID DYNAMICS RESEARCH LABORATORY

Grant No. AF AFOSR 156-64

Final Scientific Report

June 1964

Prepared for

OFFICE OF SCIENTIFIC RESEARCH

UNITED STATES AIR FORCE

GRANT NO. AF AFOSR 156-64

THE CALCULATION OF TWO- AND THREE-DIMENSIONAL INVISCID UNSTEADY FLOWS BY THE METHOD OF CHARACTERISTICS

by

Harry Sauerwein, Jr.

ABSTRACT

The problem of the numerical solution by the method of characteristics of the two- and three-dimensional flow of an inviscid nonequilibrium gas is formulated. For unsteady flows the method is limited, for practical reasons, to the solution of hypersonic or other flows which have short transient times. The specific details of the finite difference network to be used and the associated finite difference equations are presented. It was found that numerical instability was possible with certain previously proposed networks. An existing stability criterion was found to apply to the multi-dimensional method of characteristics and this criterion was used to synthesize a stable finite difference network.

High speed digital computer programs are presented which perform the numerical calculation. Because the operating speed and storage capacity of the larger and faster computers available today does not allow the solution of the most general problem in a reasonable length of time, it was necessary to limit the programmed calculation to the two-dimensional unsteady flow of a perfect gas. Rather long programs which require large amounts of input data and produce even larger amounts of output data are the results of the programming effort.

Four example cases of the flow about a circular cylinder with its axis perpendicular to a Mach five free stream are presented. The cylinder is held steady, symmetrically and asymmetrically warped to an elliptic cross-section and oscillated in a direction normal to the free stream. These

cases indicate that the solution of multi-dimensional flows by the method of characteristics is both practical and feasible. With the substitution of higher order interpolation for the linear interpolation used in this initial study, accurate and useful results for engineering problems can be obtained.

The extension of the programmed procedures to the most general problem which can be solved with the method of characteristics is discussed and the steps required to extend the procedure are indicated.

ACKNOWLEDGEMENTS

The author wishes to express his appreciation to the project supervisor, Professor Holt Ashley, and to Professors Judson Baron and Garabed Zartarian for their advice and helpful criticism. The assistance of Professor Gilbert Strang in indicating the probable cause of numerical instability encountered in this work is gratefully acknowledged. Dr. P. T. Hsu also contributed useful advice and criticism.

Mark Sussman worked as a research assistant on this project. Without his assistance this project could not have been completed in a reasonable period of time.

The author also wishes to acknowledge the assistance of friends and associates in industry: Drs. Powell and Pridmore Brown at the Norair Division, Northrop Corporation; Mr. Charles Strom of the Cornell Aeronautical Laboratory; and Mr. J. F. Springfield of the Research and Advanced Development Division, Avco Corporation.

Dr. Richard Kaplan, who wrote the SCAN subroutine for this project, and Dr. Sheila Widnall assisted the author in the computer programming and debugging required for this work.

Acknowledgement is also made to the M.I.T. Computation Center for the time allowed on its IBM 7090 and later the 7094 computers, done as Problem M2413.

The production of this document is a result of the typing of Miss Patricia Beliveau and [redacted]. The figures were drawn by the M.I.T. Illustration Service.

This work was done under the auspices of the Division of Sponsored Research at the Massachusetts Institute of Technology, Project 9995, sponsored by the Air Force Office of Scientific Research, United States Air Force. This is the final report for Grant No. AF AFOSR 156-64, monitored by Capt. Lucius P. Gregg, USAF, Mechanics Division, Directorate of Engineering Sciences.

TABLE OF CONTENTS

Figures

LIST OF SYMBOLS

Symbol	Definition
a	coordinate transformation matrix
$A, B, C,$ etc.	coefficients used in various equations and defined as used
a	frozen speed of sound
a_e	equilibrium speed of sound
$\mathcal{B}$	function which defines the body surface
d	distance, defined as used
c_n	mass fraction of the nth species
F_i	general function
f	general dependent variable
h	specific enthalpy
$\vec{i}, \vec{j}, \vec{k}, \vec{l}$	unit vectors in the (x, y, z, t') directions
K	number of fluid particles which pass over the body during a typical transient motion
K_i	normalizing factor defined in Eq. (3.8)
L	typical body length
M	Mach number
N	total number of chemical species
$\vec{N}$	unit normal vector for a surface or hypersurface
$N_x, N_y, N_z, N_{t'}$	components of the unit normal vector in the (x, y, z, t') directions
p	ambient pressure
q	velocity magnitude
R	in general the constant radius of curvature of a surface, usually the radius of the cylindrical body, the gas constant only in Eqs. (4.3) and (4.4)

s	specific entropy
T	temperature
t	time
t'	transformed time coordinate with a dimension of length, see Eq. (2.2)
U	arbitrary constant with a dimension of velocity used to transform the time coordinate, see Eq. (2.2)
u	velocity component in the x-direction
$\vec{V}$	generalized velocity vector defined by Eq. (2.7)
v	velocity component in the y-direction
w	velocity component in the z-direction
X	distance from the particle line base point to the new point, given by Eq. (3.16c)
x, y, z	Cartesian coordinates
Z	distance from the base point to the new point defined in Eq. (3.10e)
α	angle of attack
α_n	general coordinates
β_i	characteristic coordinates
γ	ratio of specific heats
Δ	change in quantity
ϵ	defined by Eq. (5.4)
θ	angle in a coordinate transformation
λ_i	linear coefficients in the compatibility equation
μ	Mach angle
μ_n	chemical potential of the nth species
ξ	coordinate along the particle line
ρ	mass density
σ_n	chemical source function for the nth species

τ	total number of time steps in a calculation
Φ	defined by Eq. (3.26b)
ϕ	angle in a coordinate transformation
Ψ	angle in a coordinate transformation
ω	frequency of oscillation
$(\ )_{ave}$	average quantity
$(\ )_B$	refers to the body surface
$(\ )_c$	central point
$(\ )_i$	Base point property or coordinate in the finite difference network
$(\ )_N$	normal component
$(\ )_p$	refers to a plane
$(\ )_S$	refers to the shock wave
$(\ )_{sph}$	refers to a sphere
$(\ )_T$	tangential component
$(\ )_+$	condition in front of the shock wave
$(\ )_-$	condition behind the shock wave
$(\ )_\infty$	free stream condition
$(\ )^{(n)}$	quantity at the nth iteration step

CHAPTER 1

INTRODUCTION

This study was initiated with the purpose of analyzing inviscid hypersonic flow about three-dimensional bodies in unsteady motion. In the course of this investigation, a more general method has been considered which applies to all inviscid unsteady flows, not just hypersonic flows. There are practical limitations on the application of the method, however.

The study of hypersonic flow is mathematically nonlinear in most cases, perhaps by definition. A review of the literature reveals that because of this nonlinearity, theoretical study is very difficult. Many approximations are usually made so that the mathematical problem becomes amenable to analytical methods. See, for example, the book by Hayes and Probstein[1]. Many times it is very difficult to evaluate the effect of the approximations on the results of the analysis after the approximate solution has been obtained. When unsteady hypersonic flow is considered, the problems are even more challenging. Here, by unsteady hypersonic flow, we mean truly unsteady flow where the "quasi-steady" approximation which simplifies certain unsteady hypersonic analysis is not valid. Also, without exact solutions against which these approximate theories can be checked, the

- - - - - - - - - - - - - - - - -

[1] Superscripts refer to the reference numbers listed in the section entitled References.

errors introduced by them are very difficult, if not impossible to determine. Hence, a means of exactly calculating unsteady hypersonic flow is very desirable.

The question which faces us then is, what is the best method for determining exact solutions to unsteady hypersonic flows? A method is sought to solve the general nonlinear equations which describe the flow of an inviscid nonequilibrium gas. It is probably safe to assume that a general analytical solution can not be found because of the complicated nature of the equations. It must be concluded then, that the only hope lies in the direction of a numerical solution.

The general, unsteady flow of a gas is governed by a set of hyperbolic, quasi-linear partial differential equations. In general, there are two methods for numerically solving such systems of equations. One method replaces the derivatives in the original set of equations with finite differences, and seeks to solve numerically, this set of difference equations. This approach, we might call the finite difference method. A second procedure first writes the equations in characteristic form to somewhat simplify them and then utilizes finite differences to approximate the derivatives. The numerical solution is obtained by integration in the characteristic surfaces, hence, this approach is referred to as the method of characteristics. A discussion and examples of these two basic approaches are given in Chapters 15 and 16 of the book edited by Ralston and Wilf[2].

The finite difference methods might be further subdivided into standard methods and methods utilizing artificial viscosity. In the latter approach, an extra artificial viscosity term (or in some cases, the real viscosity term) is added to the equations such that free boundaries, e.g. shock waves, are automatically determined in the solution. This approach has been studied by von Neumann and Richtmyer[3], Lax[4], and Lax and Wendroff[5] among others. Thus, in general there are three basic numerical approaches which might be utilized in attacking the problem. Hartree[6] has proposed a method which combines the standard finite difference approach with the method of characteristics mainly to obtain the solution on an evenly spaced grid while still taking advantage of the characteristics. Such hybrid approaches usually have some of the disadvantages of the original methods and quite often require more computations than the original methods, so that they are not considered here.

The advantages and disadvantages of each of the three basic approaches are compared in the following table. Some of the properties given in the table were obtained from the books by Fox[7] (Chapters 17, 18, 26, 27 and 28), Forsyth and Wasow[8] (parts 1 and 4), Richtmyer[9] and Collatz[10] (Chapter IV, Part 5) while some of the estimates are the author's.

Comparison of Possible Methods of Solution

Standard Finite Difference Approach	Finite Difference Approach with Artificial Viscosity	Method of Characteristics
1. How are free boundaries handled?		
specially treated as discontinuities	automatically handled with artificial viscosity and by retaining conservative properties in the difference equations	specially treated as discontinuities
2. How are fixed boundaries handled?		
specially treated	specially treated, but some difficulty can arise because the artificial viscosity increases the order of the equations and extra boundary conditions are available which are not easy to specify in some cases	specially treated
3. What formulation of the equations is usually preferred?		
Eulerian or Lagrangian	Lagrangian in some problems	Eulerian, usually utilized
4. Is a stability criterion required?		
yes	yes	yes, but it is usually simplified by knowledge of the location of the characteristic surfaces
5. How many independent variables can be treated?		
three space dimensions and time	more suited to two independent variables, but can be generalized	three space dimensions and time

Standard Finite Difference Approach	Finite Difference Approach with Artificial Viscosity	Method of Characteristics

6. How difficult is the machine programming task?

somewhat complicated because boundaries must be treated specially	simplest	most difficult

7. How much computing time is required?

about the same as finite difference with artificial viscosity	about the same as standard finite difference approach	potentially the quickest method

8. How much machine storage is required?

about the same as finite difference with artificial viscosity	about the same as standard finite difference approach	slightly more than the other two methods

9. On what sort of grid is the solution obtained?

evenly spaced	evenly spaced	characteristic grid, probably will require interpolation of the final result

10. What is the accuracy of the results?

intermediate accuracy	least accurate in that the free boundaries are not precisely located and errors can be introduced by not properly specifying the extra boundary conditions introduced by the artificial viscosity	potentially the most accurate, also this method of solution most closely follows the physical model

From the estimates and properties given in the above table, it is possible to draw some conclusions about the utility of applying the methods to various types of problems. The first property listed in the table shows a basic difference between the approach utilizing artificial viscosity and the other two approaches. This is the major advantage of the artificial viscosity method, but it is at the expense of points 2 and 10. Hence, if the problem to be solved involved a complicated interaction of shock waves, for example, this might be the best method to utilize. However, if only simple shock wave configurations were to be considered, perhaps one of the other two methods might best be utilized. On points 1, 2, 3, and 5 the standard finite difference approach (SFDA) is similar to the method of characteristics (MOC), while on points 6, 8, and 9 the SFDA seems to have the advantage, and on points 4, 7, and 10 the MOC seems to be preferred.

The approach to be adopted here must take into account the type of problems that are to be solved. In this study, we are interested generally in determining the external hypersonic flow about bodies performing unsteady motions. At least initially, and in a majority of the problems to be considered, the bodies are simple enough that rather simple shock configurations are encountered. Hence, the approach utilizing artificial viscosity is not required. The choice between the SFDA and the MOC is not as simple. As has been indicated, the SFDA has advantages on three points and the MOC has three, while four can be considered

equal. Thus, the individual points must be considered to choose the better method of approach. The SFDA is favored by points 6, 8, and 9, which mainly apply to the ease of writing the program and applying the method to a computing machine. The MOC, on the other hand, is favored by points 4, 7, and 10 which are advantages of the finished working method. So, if ease of writing the program and initially applying the method is most important, it would appear that the SFDA is better, but if extra initial effort can be devoted to programming, a better working method would seem to result from the MOC. It is felt that, because once a machine program is written it can be used over and over to solve many problems, it is worth an extra initial effort to write a more accurate and quicker program. Hence, the MOC is adopted in this study. Note that this choice is based on some rather crude estimates and the validity of these conclusions ultimately, can be judged only after working programs have been written and are compared.

The MOC was first proposed by Massau[11] in 1899 and is sometimes referred to as the Method of Massau. The method was adopted to solve hyperbolic problems in fluid mechanics, perhaps first in 1929 by Prandtl and Busemann[12]. The MOC was first used to solve two-dimensional irrotational supersonic steady flows and one-dimensional unsteady flows of a perfect gas. These calculations could be made by hand and in part graphically. Necessarily, the first applications of the method had to be limited to simplified problems in two independent variables, because of the

impossibility of doing more ambitious problems with hand calculations. In the 1940's and early 1950's, when electronic digital computers first became available, the MOC was developed extensively for more general problems involving two independent variables. Ferri[13] and Meyer[14,15] discuss this work and give an extensive list of references. The rotational flows of real gases became amenable to solution by the MOC when computers were used and in some cases, were merely a matter of the routine application of general computer programs. In the late 1950's and early 1960's, computer development had increased the storage capacity and speed of operation of the machines to the point that more ambitious problems could be attempted with the MOC. Problems involving three independent variables and nonequilibrium thermodynamics are now being attempted for the first time. These new advances in computer technology have made it possible to at least consider the most general types of problems which can be solved with the MOC.

Briefly, the MOC is used to solve Cauchy problems for a set of quasi-linear, hyperbolic partial differential equations. Data are given on an initial surface* together with boundary conditions on the flow field, such as solid body surfaces and shock waves. The solution is then obtained on an adjacent surface by

- - - - - - - - - - - - - - - - - -

* The terminology of a problem involving three independent variables is used here to emphasize that problems of more than two independent variables can be handled while still avoiding the more general, but perhaps confusing terminology of four independent variables, e.g. hypersurfaces.

numerical solution of the equations written in characteristic form with the partial derivatives replaced by finite differences. The solution on this new surface is then taken as the initial data and the process repeated to obtain the solution on the next adjacent surface. In this way, the solution is obtained by marching in space or time until the desired solution is obtained or until the complete region specified by the initial data has been determined.

A review of the literature of the MOC for two independent variables cannot be given here, as the volume of such literature is enormous. An attempt will be made here to list only the most recent work on the MOC which involves problems which include more than two independent variables and those that consider nonequilibrium thermodynamics.

The work on problems with more than two independent variables can be divided into two groups. One group, theoretically studies the problems and formulates them, and in some cases even goes so far as to write the finite difference equations, while the second group includes work which has gone on to actually do some numerical calculations on digital computers. The former group is by far the larger of the two groups, while the latter has less than ten members to be sure. The reason for this, is that much time and effort are required to carry a formulation through to a working computer program or to accomplish the calculation by hand in simple cases. Thus, many studies of the MOC have been terminated after formulating the

problem. Included in the first group are the basic works of Thornhill[16] and Coburn and Dolph[17]. Sauer[18], Clippinger and Giese[19] and Holt[20] have also made contributions to the theory. Coburn[21] and Saini[22] have used tensor analysis to formulate a much more general problem which has the particular problem considered here as a special case. Most recently, Fowell[23] and Sauer[24] have published work in this area. In the second group of papers, which did some actual numerical calculations, the first few studies were done when computers were not readily available so that they involved simplified calculations which could be accomplished by hand (by hand calculations, we also mean with the aid of desk calculators). These initial calculations include the work of Moeckel[25], Ferrari[26], and Bruhn and Haack[27]. Later work by Butler[28] and Tsung[29] was accomplished using digital computers. Results obtained with the latest generation of computers have just begun to appear. Morretti et.al.[30] have published some recent results. More and more actual calculations are being carried out at the present time on more and more ambitious problems because of the increased versatility of digital computers.

The study of hypersonic flows must allow for the possibility of high temperature real-gas effects and nonequilibrium thermodynamics caused by the high temperatures, and perhaps low densities which can be encountered in practical flow fields. Recent work in this areas has seen the incorporation of real-gas equilibrium thermodynamics in problems with two independent

variables in an almost routine manner. Initial work has begun on including nonequilibrium thermodynamics in two independent variable problems. The papers by Chu[31] and Broer[32] discuss the formulation of the problem, while the papers by Sedney et.al.[33,34], Capiaux and Washington[35], and Wood et.al.[36] also give some initial results of numerical calculations. To the author's knowledge, no work has been published on problems involving more than two independent variables together with nonequilibrium thermodynamics.

The object of this work, therefore, is to formulate the method of characteristics for general three-dimensional unsteady flow problems, allowing for the possibility of nonequilibrium thermodynamics. It is also the purpose of this work to write and check a working computer program which applies the method to practical flow problems. It may be impossible or impractical at this time to write a completely general program, but only those compromises and approximations which are absolutely necessary to the writing of a working program will be made. In writing the program, it will always be borne in mind that, even though some compromises must be made, they will eventually be removed at a future time. In this way, the compromises and approximations can be introduced in such a way that they can be more easily removed later.

CHAPTER 2

THE EQUATIONS FOR THE GENERAL METHOD

The quasi-linear, first-order partial differential equations for the flow of an inviscid nonequilibrium gas are set forth in this chapter. A brief review of the theory of characteristics is presented in Appendix A. The theory is used in this chapter to derive the characteristic and compatibility equations which are required for the method of characteristics.

2.1 The Equations of Change

The equations which govern the flow of a nonequilibrium gas are the conservation laws, together with the equation of state. In cartesian coordinates (x, y, z, t) they have the following form.

continuity

$$\frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial x} + v\frac{\partial \rho}{\partial y} + w\frac{\partial \rho}{\partial z} + \rho\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right) = 0 \tag{2.1a}$$

momentum conservation in x-direction

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} + \frac{1}{\rho}\frac{\partial p}{\partial x} = 0 \tag{2.1b}$$

momentum conservation in y-direction

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} + \frac{1}{\rho}\frac{\partial p}{\partial y} = 0 \qquad (2.1c)$$

momentum conservation in z-direction

$$\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} + \frac{1}{\rho}\frac{\partial p}{\partial z} = 0 \qquad (2.1d)$$

energy conservation

$$\frac{\partial h}{\partial t} + u\frac{\partial h}{\partial x} + v\frac{\partial h}{\partial y} + w\frac{\partial h}{\partial z} - \frac{1}{\rho}\left(\frac{\partial p}{\partial t} + u\frac{\partial p}{\partial x} + v\frac{\partial p}{\partial y} + w\frac{\partial p}{\partial z}\right) = 0 \qquad (2.1e)$$

continuity of nth species

$$\frac{\partial c_n}{\partial t} + u\frac{\partial c_n}{\partial x} + v\frac{\partial c_n}{\partial y} + w\frac{\partial c_n}{\partial z} - \frac{1}{\rho}\sigma_n = 0 \qquad (n = 1, 2, \ldots, N) \qquad (2.1f)$$

state equation

$$h = h(p, \rho, c_1, \ldots, c_N) \qquad (2.1g)$$

Where in general

$$\sigma_n = \sigma_n(p, \rho, c_1, \ldots, c_N) \qquad (n = 1, 2, \ldots, N) \qquad (2.1h)$$

is a given function determined from the chemistry of the gas being considered. The chemical source function, σ_n, is a complicated function of the chemical reactions and reaction rates being considered[39]. These are 2N + 6 equations in 2N + 6

unknowns and thus form a complete set.

It should be noted that the species continuity equation (2.1f) which is sometimes called the rate equation is really more general than it might appear above. Vibrational relaxation and element ionization are also described by equations with the same form as (2.1f). These equations are obtained by utilizing vibrational energy or electron concentration instead of the species mass fractions and by using the proper source function, σ .

In order to eliminate the two algebraic equations (2.1g-h) they must be substituted into equations (2.1e) and (2.1f) respectively. (2.1e) then takes the form

$$\frac{Dp}{Dt} - \frac{D\rho}{Dt} a^2 + \sum_{n=1}^{N} \left\{ \frac{Dc_n}{Dt} \left[\frac{\partial h}{\partial c_n} \Big/ \left(\frac{\partial h}{\partial p} - \frac{1}{\rho} \right) \right] \right\} = 0 \qquad (2.1i)$$

where

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + u \frac{\partial}{\partial x} + v \frac{\partial}{\partial y} + w \frac{\partial}{\partial z}$$

and it is understood that all other thermodynamic variables are held constant when the enthalpy, h, is partially differentiated with respect to any one thermodynamic variable. a is the speed of sound in the gas with the mass fractions of the species held constant, or in other words, the frozen speed of sound. It is given by

$$a^2 = -\left[\frac{\partial h}{\partial \rho} \right]_{p,c_n} \Big/ \left(\left[\frac{\partial h}{\partial p} \right]_{\rho,c_n} - \frac{1}{\rho} \right) \qquad (2.1j)$$

This is the complete set of partial differential that is to be solved.

2.2 The Characteristic Equation

As indicated in Appendix A, the characteristic equation is found by first introducing a coordinate transformation to (β_1, β_2, β_3, β_4) coordinates. It is also useful to introduce a new independent variable with the form

$$t' = Ut \tag{2.2}$$

where U is an arbitrary constant velocity used to give t' the dimension of a space coordinate. This allows the time coordinate and the space coordinates to be treated similarly. Choose the new coordinates such that

$$\beta_1 = \text{constant} \tag{2.3}$$

is a characteristic hypersurface*. The characteristic equations for the equations of change (2.1) is obtained by evaluating the determinant given in equation (A.5), Appendix A. The result is

$$\left(u\frac{\partial\beta_1}{\partial x}+v\frac{\partial\beta_1}{\partial y}+w\frac{\partial\beta_1}{\partial z}+U\frac{\partial\beta_1}{\partial t'}\right)^{N+3} \cdot\left\{\left(u\frac{\partial\beta_1}{\partial x}+v\frac{\partial\beta_1}{\partial y}+w\frac{\partial\beta_1}{\partial z}+U\frac{\partial\beta_1}{\partial t'}\right)^2 - a^2\left[\left(\frac{\partial\beta_1}{\partial x}\right)^2+\left(\frac{\partial\beta_1}{\partial y}\right)^2+\left(\frac{\partial\beta_1}{\partial z}\right)^2\right]\right\}=0 \tag{2.4}$$

* The hypersurface in this case is a three-dimensional manifold in a four-dimensional space.

From (2.4) it can be seen that there are two sets of real characteristic hypersurfaces corresponding to

$$u\frac{\partial \beta_1}{\partial x} + v\frac{\partial \beta_1}{\partial y} + w\frac{\partial \beta_1}{\partial z} + U\frac{\partial \beta_1}{\partial t'} = 0 \qquad (2.5)$$

and

$$\left(u\frac{\partial \beta_1}{\partial x} + v\frac{\partial \beta_1}{\partial y} + w\frac{\partial \beta_1}{\partial z} + U\frac{\partial \beta_1}{\partial t'}\right)^2 - a^2\left[\left(\frac{\partial \beta_1}{\partial x}\right)^2 + \left(\frac{\partial \beta_1}{\partial y}\right)^2 + \left(\frac{\partial \beta_1}{\partial z}\right)^2\right] = 0 \qquad (2.6)$$

Equation (2.5) is the dot product of the vector $\vec{V}$ (a generalized velocity vector)

$$\vec{V} = u\vec{i} + v\vec{j} + w\vec{k} + U\vec{l} \qquad (2.7)$$

where $\vec{i}$, $\vec{j}$, $\vec{k}$, $\vec{l}$ are the unit vectors along the x, y, z, t' coordinate axes respectively, with the vector $\nabla\beta_1$,

$$\nabla\beta_1 = \frac{\partial \beta_1}{\partial x}\vec{i} + \frac{\partial \beta_1}{\partial y}\vec{j} + \frac{\partial \beta_1}{\partial z}\vec{k} + \frac{\partial \beta_1}{\partial t'}\vec{l} \qquad (2.8)$$

$\nabla \beta_1$ is normal to the characteristic hypersurface. $\vec{V}$ is tangent to the particle line or world line, as it is called in relativity, (see Reference 38, p. 5, and p. 114). Therefore, hypersurfaces

made up of particle lines are characteristic. Note that the particle line is a (N + 3)-fold characteristic because the corresponding term in (2.4) is raised to the N + 3 power. This property is used to great advantage in the method of characteristics.

The second set of characteristic hypersurfaces is given by (2.6). This is a quadratic hypersurface, and is a generalization of the Mach conoid which appears in three-dimensional steady supersonic flow and two-dimensional unsteady flow (see Reference 13, pp. 642-657) and is generally referred to as the Mach hyperconoid. If the equation of the Mach hyperconoid is written in the form

$$\beta_1 = \beta_1 (x, y, z, t')$$

the differential equation which defines the hyperconoid is (2.6). Of course, this equation can not be solved in general for β_1, but at a particular point (x_i, y_i, z_i, t'_i) the hyperconoid is tangent to the local Mach hypercone and is just an envelope of sound waves emitted from the point. The equation of the Mach hypercone is easily derived from the geometry of sound waves assuming a uniform steady flow with conditions the same everywhere as at the point being considered, and is given by

$$\begin{aligned}&(x-x_i)^2+(y-y_i)^2+(z-z_i)^2\\&+(t'-t'_i)^2(u^2+v^2+w^2-a^2)\frac{1}{U^2}-2\frac{u}{U}(x-x_i)(t'-t'_i)\\&-2\frac{v}{U}(y-y_i)(t'-t'_i)-2\frac{w}{U}(z-z_i)(t'-t'_i)=0\end{aligned} \qquad (2.9)$$

The Mach hypercone can be used as a local approximation to the Mach hyperconoid.

2.3 The Compatibility Equations

The compatibility equations are obtained by forming a linear combination of the equations of change (2.1) and determining the linear coefficients such that partial derivatives in a direction normal to the characteristic surface are eliminated. The equations for the linear coefficients and compatibility condition are derived in Appendix A and given by (A.9) and (A.10). There are two sets of compatibility equations, one for each type of characteristic hypersurface. Actually, there is only one equation corresponding to the Mach hyperconoid, while there are N + 3 equations for the particle line. See Reference 38, pp. 106-107 for a discussion of the number of compatibility equations which correspond to a particular characteristic hypersurface.

The linear combination of the equations of change is given by (A.7) and has the form

$$C_k \frac{\partial u_k}{\partial \beta_1} + D_k \frac{\partial u_k}{\partial \beta_2} + E_k \frac{\partial u_k}{\partial \beta_3} + F_k \frac{\partial u_k}{\partial \beta_4} + G = 0 \qquad (2.10)$$

For the given equations of change, (2.1), the compatibility equation for the Mach hyperconoid has the following terms:

$$\Psi_1 = \lambda_1 \rho \frac{\partial \beta_m}{\partial x} + \lambda_2 (\vec{v} \cdot \nabla \beta_m) \qquad (2.11a)$$

$$\Psi_2 = \lambda_1 \rho \frac{\partial \beta_m}{\partial y} + \lambda_3 (\vec{V} \cdot \nabla \beta_m) \qquad (2.11b)$$

$$\Psi_3 = \lambda_1 \rho \frac{\partial \beta_m}{\partial z} + \lambda_4 (\vec{V} \cdot \nabla \beta_m) \qquad (2.11c)$$

$$\Psi_4 = (\lambda_1 + \lambda_5)(\vec{V} \cdot \nabla \beta_m) \qquad (2.11d)$$

$$\Psi_5 = \frac{1}{\rho}\left(\lambda_2 \frac{\partial \beta_m}{\partial x} + \lambda_5 \frac{\partial \beta_m}{\partial y} + \lambda_4 \frac{\partial \beta_m}{\partial z}\right) - \frac{\lambda_5}{a^2}(\vec{V} \cdot \nabla \beta_m) \qquad (2.11e)$$

$$\Psi_{5+n} = \left\{ \lambda_{5+n} + \lambda_5 \left[\frac{\partial h}{\partial c_n} \Big/ \frac{\partial h}{\partial \rho} \right] \right\} (\vec{V} \cdot \nabla \beta_m) \qquad (2.11f)$$

$$(n = 1, \ldots, N)$$

where

$$\vec{V} \cdot \nabla \beta_m = u \frac{\partial \beta_m}{\partial x} + v \frac{\partial \beta_m}{\partial y} + w \frac{\partial \beta_m}{\partial z} + U \frac{\partial \beta_m}{\partial t'} \qquad (2.12)$$

and

$\Psi_k = C_k$ when m = 1

$\Psi_k = D_k$ when m = 2

$\Psi_k = E_k$ when m = 3

$\Psi_k = F_k$ when m = 4

also

$$G = \frac{1}{\rho} \sum_{n=1}^{N} \lambda_{5+n} \sigma_n \qquad (2.13)$$

These equations correspond to equations (A.8) in Appendix A. The linear coefficients, λ_k, are determined by setting C_k equal to zero and solving the resulting set of equations.

This eliminates derivatives in a direction normal to the characteristic hypersurface. The result is

$$\lambda_2/\lambda_1 = -\rho \frac{\partial \beta_1}{\partial x} / (\vec{V} \cdot \nabla \beta_1) \tag{2.14a}$$

$$\lambda_3/\lambda_1 = -\rho \frac{\partial \beta_1}{\partial y} / (\vec{V} \cdot \nabla \beta_1) \tag{2.14b}$$

$$\lambda_4/\lambda_1 = -\rho \frac{\partial \beta_1}{\partial z} / (\vec{V} \cdot \nabla \beta_1) \tag{2.14c}$$

$$\lambda_5/\lambda_1 = -1 \tag{2.14d}$$

$$\lambda_{5+n}/\lambda_1 = \frac{\partial h}{\partial c_n} / \frac{\partial h}{\partial \rho}, \qquad (n = 1, \ldots, N) \tag{2.14e}$$

when equations (2.11) through (2.14) are substituted into (2.10), the following compatibility equation for the Mach hyperconoid results.

$$\begin{aligned}
\sum_{m=2}^{4} \Bigg\{ & \frac{\partial u}{\partial \beta_m} \left[\frac{\partial \beta_m}{\partial x} (\vec{V} \cdot \nabla \beta_1) - \frac{\partial \beta_1}{\partial x} (\vec{V} \cdot \nabla \beta_m) \right] \\
& + \frac{\partial v}{\partial \beta_m} \left[\frac{\partial \beta_m}{\partial y} (\vec{V} \cdot \nabla \beta_1) - \frac{\partial \beta_1}{\partial y} (\vec{V} \cdot \nabla \beta_m) \right] \\
& + \frac{\partial w}{\partial \beta_m} \left[\frac{\partial \beta_m}{\partial z} (\vec{V} \cdot \nabla \beta_1) - \frac{\partial \beta_1}{\partial z} (\vec{V} \cdot \nabla \beta_m) \right] \\
& + \frac{\partial p}{\partial \beta_m} \frac{1}{\rho} \Bigg[\frac{1}{a^2} (\vec{V} \cdot \nabla \beta_m)(\vec{V} \cdot \nabla \beta_1) \\
& \qquad - \left(\frac{\partial \beta_m}{\partial x} \frac{\partial \beta_1}{\partial x} + \frac{\partial \beta_m}{\partial y} \frac{\partial \beta_1}{\partial y} + \frac{\partial \beta_m}{\partial z} \frac{\partial \beta_1}{\partial z} \right) \Bigg] \Bigg\} \\
& + \left[(\vec{V} \cdot \nabla \beta_1) / \left(\rho \frac{\partial h}{\partial \rho} \right) \right] \left[\sum_{n=1}^{N} \frac{\partial h}{\partial c_n} \sigma_n \right] = 0
\end{aligned} \tag{2.15}$$

where

$$\beta_1 = \text{constant}$$

is the characteristic Mach hyperconoid. Note that derivatives of ρ and c_n do not appear in the equation.

As mentioned above, there are N + 3 compatibility equations for the particle line, but only N + 1 of these equations are needed for the method of characteristics. These could be obtained in a manner similar to the procedure just used to obtain equation (2.15). However, because the equations sought do not have derivatives in a direction normal to the particle line and thus involve derivatives along the particle line only, it is simpler to use this property to choose them by inspection from the original partial differential equations.

The substantial derivative

$$\frac{D}{Dt} = u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y} + w\frac{\partial}{\partial z} + U\frac{\partial}{\partial t'}$$

is the derivative of a dependent variable following a fluid particle, so if ξ is taken as a coordinate along the particle line in the (x, y, z, t') space, equations (2.1f) can be written

$$\rho V \frac{dc_n}{d\xi} = \sigma_n \qquad (n = 1, \ldots, N) \qquad (2.16)$$

where

$$V = \sqrt{u^2 + v^2 + w^2 + U^2}$$

These are N of the compatibility equations along the particle line. One additional relation is needed. The energy equation, which can be written as (2.1f) or as (2.1i), meets this need. The energy equation can be written in a somewhat simpler form if the specific entropy is introduced. The entropy is a function of the thermodynamic variables and the mass fractions of species

$$s = s(p, \rho, c_1, \ldots, c_N) \tag{2.17}$$

so

$$\frac{Dp}{Dt} = \frac{Ds}{Dt}\left(\frac{\partial p}{\partial s}\right)_{\rho,c_n} + \frac{D\rho}{Dt}\left(\frac{\partial p}{\partial \rho}\right)_{s,c_n} + \sum_{n=1}^{N} \frac{Dc_n}{Dt}\left(\frac{\partial p}{\partial c_n}\right)_{s,\rho,c_{j\neq n}}$$

where the subscripts indicate the variables held constant in the partial differentiations. Substituting for Dp/Dt in (2.1i) and utilizing another form of the expression for the frozen speed of sound

$$a^2 = \left(\frac{\partial p}{\partial \rho}\right)_{s,c_n} \tag{2.18a}$$

the following is obtained

$$\left(\frac{\partial p}{\partial s}\right)_{\rho,c_n} \frac{Ds}{Dt} = -\sum_{n=1}^{N}\left\{\frac{Dc_n}{Dt}\left[\left(\frac{\partial p}{\partial c_n}\right)_{s,\rho,c_{j\neq n}} + \frac{\left(\frac{\partial h}{\partial c_n}\right)_{p,\rho,c_{j\neq n}}}{\left(\frac{\partial h}{\partial p}\right)_{\rho,c_n} - \frac{1}{\rho}}\right]\right\} \tag{2.18b}$$

Substituting for Dc_n/Dt from (2.1f) and introducing the ξ coordinate gives

$$\rho V\left(\frac{\partial p}{\partial s}\right)_{\rho,c_n} \frac{ds}{d\xi} = -\sum_{n=1}^{N}\left\{\sigma_n\left[\left(\frac{\partial p}{\partial c_n}\right)_{s,\rho,c_{j\neq n}} + \frac{\left(\frac{\partial h}{\partial c_n}\right)_{p,\rho,c_{j\neq n}}}{\left(\frac{\partial h}{\partial p}\right)_{\rho,c_n} - \frac{1}{\rho}}\right]\right\} \tag{2.18c}$$

Equation (2.18c) can also be written in terms of the chemical potential of the species, μ_n, and the temperature, T, (see Reference 39, equation (21)).

$$\rho T V \frac{ds}{d\xi} = - \sum_{n=1}^{N} (\mu_n \sigma_n) \qquad (2.18d)$$

Equations (2.16) and (2.18c) are the required compatibility relations for the particle line.

2.4 The Nonequilibrium Terms

As was mentioned above in Section 2.1, the chemical source function is a complicated function of the exact chemical reactions being considered and their reaction rates. The reaction rates, in turn, can be a function of temperature. The thermodynamic functions

$$h = h\ (p,\ \rho\ ,\ c_1,\ \ldots.,\ c_N)$$
$$s = s\ (p,\ \rho\ ,\ c_1,\ \ldots.,\ c_N)$$

can also be complicated functions of the mass fractions. Specific forms of these various functions are not considered here as they vary with the type of reactions being considered. The general forms of the reaction rates in certain simple cases can be obtained analytically, but in practice for most reactions, they must be determined experimentally. Because of this, many of the functional relationships are unknown, or at least uncertain at the present time. Therefore, only

general functional forms are used here. This in no way limits the results of the method of characteristics formulated here. When a specific case is being considered, it is merely a matter of substituting the proper functional relations in the appropriate equations in order to obtain the governing partial differential equations. Note that we are referring to the partial differential equations here. The solution of the equivalent finite difference equations is by no means simple, as is pointed out in Section 3.2 of the next chapter.

CHAPTER 3

THE ELEMENTS OF THE INTEGRATION PROCEDURE

In this chapter, the partial differential equations developed in Chapter 2 are written in finite difference form and the numerical procedures for solving the resulting equations are presented. The process involves choosing a finite difference network and then developing the finite difference equations which correspond to it.

3.1 The Choice of the Network Configuration

The method of characteristics for two independent variables has been used extensively and involves rather simple geometrical principles. In considering more than two independent variables, the geometry of the characteristic hypersurfaces becomes more complicated and some fundamental changes have to be made to extend the method to multi-variable problems. First, we shall consider the extensions required to go from two to three independent variables after which the extensions to four variables can be more easily determined.

Consider the geometry of the characteristic lines for two independent variables as shown in Figure 1. If the initial data are given on the line P_1P_2 at points P_1 and P_2, there is only one possible characteristic net available to determine point P_3. This is because only two characteristic lines pass

through each point and the only possible adjacent intersection on the right side on the initial data line is at point P_3. Introducing the streamline or particle line, which is also a characteristic line, does not allow other network configuration either. Hence, in obtaining the solution by marching to the right from the initial data line P_1P_2 there is only one possible characteristic net which can be used.

This is not the case in problems involving three independent variables. When the extra variable is added, an extra "degree of freedom" enters which allows a choice in the net configuration. This can best be seen by referring to Figure 2. One sheet of the Mach conoid (e.g. in the downstream or positive time direction) through a point P is depicted. This conoid is a special case of the second characteristic surface given by (2.6), but with one of the independent variables omitted. This Mach conoid is just the envelope of all the local infinitesimal Mach cones. As shown in Figure 3, the local Mach cones are tangent to the Mach conoid at every point on its surface. A line on the Mach conoid which is everywhere tangent to the generators of the local Mach cones, is termed bicharacteristic and is shown in Figure 2. It is seen that there is a single parameter family of bicharacteristics on the conoid surface. The parameter defining a bicharacteristic might be chosen as the angle θ shown in Figure 2.

Next, it is useful to introduce the concept of characteristic surfaces corresponding to a curve. Consider a specified

curve PQ shown in Figure 4 together with the conoids corresponding to each point on the curve. The envelopes of the upper and lower portions of the conoids are the characteristic surfaces corresponding to the curve PQ. Note that the lines of tangency of these surfaces with the conoids are the bicharacteristics.

Now, returning to Figure 2, consider the normal vector $\vec{N}$ which is normal to the conoid surface at P and hence, is normal to the tangent to the bicharacteristic at P. $\vec{N}$ is also normal to one of two characteristic surfaces corresponding to a line through P and normal to $\vec{N}$. Thus for each normal vector and bicharacteristic, there is at least one characteristic surface corresponding to a line through P. There is at least a single parameter family of such surfaces through every point, P. Finite difference net configurations can be made up of these surfaces and because of the availability of an infinite number of such surfaces through each point, there is an additional "degree of freedom" in the choice of the net.

It is a simple extension to reason that when a fourth independent variable is included, a second degree of freedom in the choice of the net configuration is added. This is because, corresponding to the characteristic surfaces, characteristic hypersurfaces (three-dimensional manifolds in the four-space) could be obtained at each point, and a two parameter family of such hypersurfaces would be available. There would then be a double infinity of hypersurfaces to choose from in making up a net configuration.

With this understanding of the generalization of the finite difference net to problems involving three and four independent variables, the choice of a specific net configuration can now be attempted. Again, consider the three independent variable case before trying the more complicated four variable problem. Various network configurations can be proposed. Fowell[23] has reviewed several networks which have been proposed and in some cases, utilized by various people. We will not go into the details of the networks here, but will give a very brief description of each and compare their advantages and disadvantages. For further details, the reader is referred to Fowell and the original papers which first described each network.

Thornhill[16] first proposed two network configurations which Fowell has termed the tetrahedral characteristic line network and the tetrahadral characteristic surface network. The nomenclature introduced by Fowell is adopted here. These networks are shown in Figures 5 and 7. A minimum of three points is required in the initial surface to calculate conditions at a new point off the surface. In the tetrahedral characteristic line network, three Mach cones through the three initial points, P_1, P_2, and P_3, are utilized to form the net. A mutual intersection point of the three cones is the new point at which the flow properties are to be calculated. The generators of the cones which pass through the intersection point are approximations to bicharacteristics. The compatibility equations can be written in finite difference form along these bicharacteristics and then

utilized to determine the flow properties at the new point. If this network is considered from the standpoint of numerical stability, however, it is found to be unstable (see Appendix B). For this reason, this writer proposes the modified tetrahedral characteristic line network shown in Figure 6.

In the modified network, it is proposed that the triangle formed by the three initial points be inscribed with a circle, and the points of tangency of the circle with the triangle, P_{12}, P_{23}, and P_{31}, be utilized in exactly the same way as the initial points, P_1, P_2, and P_3, for the unmodified network. The properties at the points P_{12}, P_{23}, and P_{31} might be obtained by linear interpolation between the initial points or a higher order interpolation scheme might be utilized in the initial surface. The addition of the procedure of inscribing the circle within the triangle assures numerical stability for this network (see Appendix B).

The tetrahedral characteristic surface network also uses the minimum of three points in the initial surface to form the net. The lines passing through each pair of points are used to define characteristic surfaces. The mutual intersection point of the three inward leaning characteristic surfaces is utilized as the new point, P_4. The characteristic conoid through P_4 opening toward the initial surface is tangent to the characteristic surfaces along bicharacteristics as shown in Figure 7. The points P_{12}, P_{23}, and P_{31} at the base of the bicharacteristics in the initial surface are utilized in the finite difference

equations and the flow properties at these points must be determined by interpolation in the initial surface.

Ferrari[26] and Sauer[18] have proposed a network made up of characteristic surfaces and two orthogonal families of reference planes as shown in Figure 8. This network of intersections of reference planes with characteristic surfaces utilizes two families of coordinate planes (e.g. x = constant and z = constant) as depicted in Figure 8. The points P_1, P_2, P_3 and P_4 in the initial surface are used to calculate the flow at the new points P_5 and P_6 which are located at the intersections of a "left running" characteristic surface from P_1P_2, a "right running" characteristic surface from P_3P_4 and a specified x = K coordinate plane. The flow properties at P_5, for instance, are obtained from equations for the variations of flow properties along the lines P_1P_5 and P_3P_5 in the characteristic surface. The third equation is obtained by considering the variation in the initial surface (along P_1P_2 or P_3P_4) and relating it to the variations along P_1P_5 and P_3P_5. The flow properties at P_6 are obtained similarly, but note that the base point P_4 will, in general, have to be moved in an iteration to P_8 in order to cause P_6 to fall on the x = K coordinate plane. This will require interpolation or extrapolation in the initial surface in order to find the flow properties at P_8.

The prismatic network of characteristic surfaces was introduced by Coburn and Dolph[17] and refined by Holt[20]. It is shown in Figure 9. This network is similar to the network of

intersections of reference planes with characteristic surfaces except that bicharacteristics are used in integrating in the characteristic surface rather than the intersections with orthogonal coordinate planes. This network also requires that the flow be known on a surface other than the initial surface such as surface $P_2P_4P_6$ in the figure. This might be a plane of symmetry, but it seems to limit the useful application of the network. The flow is determined at the new point P_5 by solving equations written along the lines P_1P_5, P_3P_5 and P_6P_5.

Butler[28] has proposed a network and method which is not precisely the method of characteristics, but is more of a combination of the standard finite difference approach and the method of characteristics in the spirit of the method of Hartree[6] mentioned in Chapter 1. Fowell termed this the pentahedral bicharacteristic line network. As shown in Figure 10, the coordinates of the new point P_0 are chosen, and the Mach cone is projected back toward the initial surface. Four bicharacteristics, one more than the minimum number required, are used in determining the flow properties. The addition of the fourth equation allows the elimination of the partial derivatives of the flow properties at the new point. The flow properties at base points, P_1, P_2, P_3 and P_4, which lie in the initial surface, must be obtained by interpolation. Note that the location of the bicharacteristics is not arbitrary, but is determined by the condition for the elimination of the partial derivatives so that the simplification of eliminating some

derivatives is traded for the complication in determining the location of the bicharacteristics and the addition of an extra equation.

Sauer[24] and Schaetz[40] have proposed methods which utilize what they call "near Characteristics", but which are very similar to the prismatic network of characteristic surfaces and the network of intersections of reference planes with characteristic surfaces discussed above. It is felt that the near characteristic networks have advantages and disadvantages similar to those discussed above for the prismatic and reference plane networks.

Fowell has discussed in detail, the advantages and drawbacks of each of the networks outlined above. We only summarize here, his discussion adding a point about stability of which he was not aware. In general, three points can be used to compare the networks. They are: (1) Is interpolation required in the initial surface? This is to be avoided because it can introduce inaccuracy. (2) Do the base points in the initial surface move about as the solution is iterated to higher order in the step size? If the points do move, interpolation is required. (3) Are the equations to be solved, written along bicharacteristic curves? The formal proofs of existence and uniqueness given by Titt[41] apply only for integration along bicharacteristics.

Interpolation is required for all the networks except the tetrahedral characteristic line network. Unfortunately, it is numerically unstable. The modified tetrahedral characteristic

line network requires only one interpolation so that it is the next best. The others require considerably more interpolation.

The base points must be moved for the last three networks outlined above and shown in Figures 8, 9, and 10. They are fixed for the first three networks.

Finally, on the third point, bicharacteristics are used in all but the network of intersections of reference planes with characteristic surfaces. It should be noted, however, that Morretti et.al.[30] have used this net and have been successful in obtaining practical results so that the argument for using bicharacteristics is not clearly established. It should also be noted that the new points are obtained on equally spaced coordinate surfaces using this net so that boundary conditions can be more easily handled in some problems.

There are other advantages and disadvantages of these network configurations that cannot be presented here because of space limitations, but the major points have been discussed above. Fowell felt that the tetrahedral characteristic line network was the most favorable and, at least in a relative sense, that its advantages outweighed its disadvantages. With the discovery of the numerical instability in this network, this writer believes that the modified tetrahedral characteristic line network is now the favored approach. Admittedly, this sort of reasoning is very qualitative, but it appears to be the best that can be done short of programming all of the networks and comparing the results they give for a test problem. Of course, the

latter suggestion is out of the question practically.

Now that we have seen the difficulty in choosing a network configuration for problems involving three independent variables, it should be apparent that choosing a network for a four variable problem is even more difficult. First of all, our inability to "picture" the geometry of a four variable space complicates the choice. This does not make the task impossible, however, because the process can still be approached on a mathematical basis. It might be expected that a generalization of the modified tetrahedral characteristic line network would also be a logical, advantageous network to use in four independent variables. Proceeding with this point of view, we can devise the generalized network.

Consider four points, P_1, P_2, P_3 and P_4 lying in the initial hypersurface. Four is the minimum number of points required to determine the flow properties at the new point P_5. First, corresponding to the circle inscribed within the triangle in the three-variable networks, we inscribe a sphere within the tetrahedron formed by the initial points in the initial hypersurface. The four points of tangency of the sphere with the tetrahedron labeled P_{123}, P_{124}, P_{134} and P_{234} in the "representation" in Figure 11(a) are used as the base points for four Mach hypercones shown in Figure 11(b). The flow properties at these four points can be obtained by interpolation in the initial hypersurface. The equation of the Mach hypercones (2.9) is a second order algebraic equation in four

unknowns. This equation written for each of the four intermediate base points (i = 1, 2, 3, 4) gives four equations in four unknowns which can be solved for the coordinates of the intersection point, P_5. The compatibility equation (2.15) can be written in finite difference form along the four lines from the intermediate base points to the new intersection point. This gives four equations to determine four flow properties at the new point, for example, u, v, w, and p. The remaining flow properties at the new point, such as the entropy, s, and the species mass fractions, c_n, are determined from the compatibility equations (2.16) and (2.18c) which apply along the particle line through the new point projected back to its intersection with the initial hypersurface. The flow properties at the base of the particle line in the initial hypersurface must also be obtained by interpolation.

This completes the discussion of the network configuration to be used, but before proceeding to the details of the finite difference equations, it should be pointed out that new points on various types of boundaries must also be calculated. Points on shock waves, on body surfaces, on contact surfaces, and on other discontinuities which might arise in flow fields of interest must also be calculated. The networks described above are for what might be called field points, which are in the interior of flow fields away from the boundaries and interior discontinuities. For the flow fields of interest here, at least two other types of points must be calculated. They are points which lie on a body surface and points which lie on an

exterior shock wave surface. By exterior shock wave surface, we mean a surface beyond which the flow is known or undisturbed. Interior secondary shock waves will not be considered in this initial study, although they can be calculated with some added effort. The three types of points to be considered here, are shown in the three-dimensional representation of the four-space in Figure 12 for the flow in the nose region of a blunt body moving at a supersonic velocity.

Note that in several of the figures, three-dimensional "representations" are used for the four-dimensional space. Thus, a two-dimensional surface may be drawn in a figure, but it may represent a hypersurface and will be labeled accordingly as in Figure 12 where the body and shock hypersurface are so labeled. Surfaces depicted in the figures may indicate a surface or they may represent a hypersurface, but lines depicted in the figures always represent lines. In Figure 12, the initial hypersurface is represented by the hyperplane $t' = 0$ (the initial steady flow).

3.2 The Field Point Procedure

In this section, the details of determining the coordinates and flow properties at a new point in the field are presented. The solution of a new field point requires the completion of four steps. First, the intermediate base points must be located and the flow properties at each point determined. The location of the new field point is then determined and as the third step, the flow properties at the new point must be obtained. Finally,

a convergent iteration scheme must be established which will determine the location and flow properties at the new point to second-order accuracy in the step size. The details of each of the steps follows.

Step 1

The four base points determine a tetrahedron within which a sphere is to be inscribed. The tangency points are utilized as intermediate base points from which the Mach hypercones are projected. The flow properties at the tangency points must be obtained by interpolation.

First, the calculation is simplified by transforming to a set of cartesian coordinates (x^*, y^*, z^*, t^*) in which the t^* coordinates of the four points are equal. This is possible because the four points cannot lie in more than a three-dimensional subspace of the four-space. In this subspace, the planes which are the faces of the tetrahedron can be written in the form

$$A_i x^* + B_i y^* + C_i z^* = 1 \qquad (i = 1, \ldots, 4) \tag{3.1a}$$

The sphere has the equation

$$(x^* - x_o^*)^2 + (y^* - y_o^*)^2 + (z^* - z_o^*) = R^2 \tag{3.1b}$$

Finally, the normal vector to the sphere must be parallel to the normal to each of the faces of the tetrahedron at each point of tangency. This condition is given by the following equations.

$$A_i / |\vec{N}_p| = (x_i^* - x_o^*) / |\vec{N}_{sph}| \quad (i = 1, \ldots, 4) \tag{3.1c}$$

$$B_i / |\vec{N}_p| = (y_i^* - y_o^*) / |\vec{N}_{sph}| \quad (i = 1, \ldots, 4) \tag{3.1d}$$

where

$$|\vec{N}_p| = (A_i^2 + B_i^2 + C_i^2)^{\frac{1}{2}} \tag{3.1e}$$

$$|\vec{N}_{sph}| = \left[(x_i^* - x_o^*)^2 + (y_i^* - y_o^*)^2 + (z_i^* - z_o^*)^2\right]^{\frac{1}{2}} \tag{3.1f}$$

The third components of the normal vectors are automatically equal because of the normalization. Equations (3.1a-d) apply at each of the points of tangency, and thus these four equations written for each of the four points of tangency give 16 equations for the sixteen unknowns, x^*_i, y^*_i, z^*_i (i = 1, 2, 3, 4), x^*_0, y^*_0, z^*_0, and R.

This set of equations can be solved numerically by using a Newton iteration scheme (sometimes termed the Newton-Raphson algorithm, see Lapidus[42], p. 288). This scheme requires a first estimate of the roots of the equations. These can be taken as the center of each face of the tetrahedron for the tangency points x^*_i, y^*_i, z^*_i, the center of the tetrahedron for the center of the sphere x^*_0, x^*_0, z^*_0 and the average distance from the center of each face to the center of the tetrahedron for the radius of the sphere R.

After the coordinates of the tangency points are determined in the (x*, y*, z*, t*) coordinates, the inverse

transformation is used to return to the (x, y, z, t') system. The flow properties at the tangency points must be obtained by interpolation. The simplest procedure would be to linearly interpolate between the three corner points on each face of the tetrahedron. This could be done in the following manner.

Starting in the (x^*, y^*, z^*, t^*) coordinates for each plane face, transform to (x^+, y^+, z^+) coordinates. Drop the t^* coordinate and work in the subspace because the t^* coordinates of all points being considered are equal. Consider the face $P_1P_2P_3$ in Figure 11(a). In the transformation to the (x^+, y^+, z^+) system the origin is located at P_1 and the x^+ axis aligned with the line P_1P_2 in the following equations:

$$x^+ = [(x^* - x_1^*)\cos\theta + (z^* - z_1^*)\sin\theta]\cos\Phi + (y^* - y_1^*)\sin\phi \tag{3.2a}$$

$$y^+ = \{(y^* - y_1^*)\cos\phi - [(x^* - x_1^*)\cos\theta + (z^* - z_1^*)\sin\theta]\sin\phi\}\cos\psi + [(z^* - z_1^*)\cos\theta - (x^* - x_1^*)\sin\theta]\sin\psi \tag{3.2b}$$

$$z^+ = [(z^* - z_1^*)\cos\theta - (x^* - x_1^*)\sin\theta]\cos\psi - \{(y^* - y_1^*)\cos\phi - [(x^* - x_1^*)\cos\theta + (z^* - z_1^*)\sin\theta]\sin\phi\}\sin\psi \tag{3.2c}$$

where

$$\tan\psi = [(z_3^* - z_1^*)\cos\theta - (x_3^* - x_1^*)\sin\theta] / \{(y_3^* - y_1^*)\cos\phi - [(x_3^* - x_1^*)\cos\theta + (z_3^* - z_1^*)\sin\theta]\sin\phi\} \tag{3.2d}$$

$$\tan\phi = (y_2^* - y_1^*) / [(x_2^* - x_1^*)\cos\theta + (z_2^* - z_1^*)\sin\theta] \tag{3.2e}$$

$$\tan\theta = (z_2^* - z_1^*) / (x_2^* - x_1^*) \tag{3.2f}$$

Note that

$$x_1^+ = 0 \qquad y_1^+ = 0 \qquad z_1^+ = 0$$
$$y_2^+ = 0 \qquad z_2^+ = 0$$
$$z_3^+ = 0$$

so that P_1, P_2, P_3 and P_{123} all lie in the (x^+, y^+) plane. Interpolating in this plane with f representing a typical flow property, the property at the tangency point is given by

$$f_{123} = \frac{(f_2 - f_1)}{x_2^+} x_{123}^+ + \left[\frac{(f_3 - f_1)}{y_3^+} - \frac{(f_2 - f_1)}{x_2^+}\frac{x_3^+}{y_3^+}\right] y_{123}^+ + f_1 \qquad (3.3)$$

The coordinates and flow properties for the intermediate base points have been determined and thus the first step of the field point solution is complete.

Step 2

The coordinates of the new field point are obtained by simultaneously solving the equations of four Mach hypercones (2.9) which eminate from the four intermediate base points determined in Step 1 above. The equations can be written in the form

$$(x-x_i)^2 + (y-y_i)^2 + (z-z_i)^2 + (t'-t_i')^2(u_i^2 + v_i^2 + w_i^2 - a_i^2)\frac{1}{U^2}$$
$$-2\frac{u_i}{U}(x-x_i)(t'-t_i') - 2\frac{v_i}{U}(y-y_i)(t'-t_i') - 2\frac{w_i}{U}(z-z_i)(t'-t_i') = 0 \qquad (3.4a)$$

where (i = 1, 2, 3, 4), which designates each of the intermediate base points. These are four equations of second degree

in four unknowns and, in general, have 16 roots. Only the closest root to the initial hypersurface in the positive time direction is desired.

To solve (3.4a), consider it rewritten and expanded in the following form:

$$F_i(x,y,z,t') = F_i^{(0)} + (x-x^{(0)})\frac{\partial F_i^{(0)}}{\partial x} + (y-y^{(0)})\frac{\partial F_i^{(0)}}{\partial y}$$
$$+(z-z^{(0)})\frac{\partial F_i^{(0)}}{\partial z} + (t'-t'^{(0)})\frac{\partial F_i^{(0)}}{\partial t'} + \frac{1}{2}(x-x^{(0)})^2\frac{\partial^2 F_i^{(0)}}{\partial x^2}$$
$$+\frac{1}{2}(x-x^{(0)})(y-y^{(0)})\frac{\partial^2 F_i^{(0)}}{\partial x\partial y} + \frac{1}{2}(x-x^{(0)})(z-z^{(0)})\frac{\partial^2 F_i^{(0)}}{\partial x\partial z} \qquad (3.4b)$$
$$+ \ldots\ldots = 0$$

where

$$F_i^{(0)} = F_i(x^{(0)}, y^{(0)}, z^{(0)}, t'^{(0)})$$
$$\frac{\partial F_i^{(0)}}{\partial x} = \frac{\partial F_i}{\partial x}(x^{(0)}, y^{(0)}, z^{(0)}, t'^{(0)})$$
$$\vdots$$

and the superscript in parentheses indicates an estimate of the root or a step in an iteration process. Now, an iteration might be carried out using the following type of equations

$$x^{(n)} = x^{(n-1)} + \Delta x \qquad y^{(n)} = y^{(n-1)} + \Delta y$$
$$z^{(n)} = z^{(n-1)} + \Delta z \qquad t'^{(n)} = t'^{(n-1)} + \Delta t' \qquad (3.4c)$$

where for example

$$\Delta x = (x - x^{(0)})$$

in (3.4b). Δx can then be obtained by truncating (3.4b) at some point and solving for Δx. A first estimate of the root $(x^{(0)}, y^{(0)}, z^{(0)}, t'^{(0)})$ must be available to start the process.

Powell[23] found that, for reasonable first estimates, the second order derivatives in (3.4b) must be included to assure convergence. Powell refers to this as a second-order Newton procedure where second-order probably refers to the inclusion of second-order derivatives. Actually, the Newton algorithm includes only the first order derivatives and is accurate to second order in the error term Δx. The inclusion of the second-order derivatives is similar to the Richmond iteration procedure (see Lapidus[42], p. 292) and is good to third order in Δx.

When the second-order derivatives are included, Powell suggests iterating the following equations to obtain the correction terms.

$$\begin{aligned} F_i &+ \Delta x^{(n+1)}\left[\frac{\partial F_i}{\partial x} + \frac{\Delta y^{(n)}}{2}\frac{\partial^2 F_i}{\partial x \partial y} + \frac{\Delta z^{(n)}}{2}\frac{\partial^2 F_i}{\partial x \partial z} + \frac{\Delta t'^{(n)}}{2}\frac{\partial^2 F_i}{\partial x \partial t'}\right] \\ &+ \Delta y^{(n+1)}\left[\frac{\partial F_i}{\partial y} + \frac{\Delta x^{(n)}}{2}\frac{\partial^2 F_i}{\partial y \partial x} + \frac{\Delta z^{(n)}}{2}\frac{\partial^2 F_i}{\partial y \partial z} + \frac{\Delta t'^{(n)}}{2}\frac{\partial^2 F_i}{\partial y \partial t'}\right] \\ &+ \Delta z^{(n+1)}\left[\frac{\partial F_i}{\partial z} + \frac{\Delta x^{(n)}}{2}\frac{\partial^2 F_i}{\partial z \partial x} + \frac{\Delta y^{(n)}}{2}\frac{\partial^2 F_i}{\partial z \partial y} + \frac{\Delta t'^{(n)}}{2}\frac{\partial^2 F_i}{\partial z \partial t'}\right] \\ &+ \Delta t'^{(n+1)}\left[\frac{\partial F_i}{\partial t'} + \frac{\Delta x^{(n)}}{2}\frac{\partial^2 F_i}{\partial t' \partial x} + \frac{\Delta y^{(n)}}{2}\frac{\partial^2 F_i}{\partial t' \partial y} + \frac{\Delta z^{(n)}}{2}\frac{\partial^2 F_i}{\partial t' \partial z}\right] = 0 \end{aligned} \tag{3.4d}$$

Values of zero can be taken as first estimates of the corrections. These are substituted into the terms in brackets and the resulting linear equations can be solved for a new value of the corrections. These corrections are then substituted into the bracketed terms and the process repeated until convergence is obtained. The corrections are then used in (3.4c) to obtain new values of the coordinates.

These new coordinates are in turn used to re-evaluate F_i and its derivatives in (3.4d), and the entire process is repeated. Thus, the procedure involves a double iteration.

This procedure suggested by Fowell was utilized in the calculations carried out for this study, but it should be pointed out that, by following Richmond's iteration process more closely, one iteration required in the above double iteration can be eliminated. If (3.4b) is truncated after the first order derivatives, we obtain

$$F_i + \Delta x \frac{\partial F_i}{\partial x} + \Delta y \frac{\partial F_i}{\partial y} + \Delta z \frac{\partial F_i}{\partial z} + \Delta t' \frac{\partial F_i}{\partial t'} = 0 \qquad (3.4e)$$

These equations can be solved for the corrections, and used to eliminate the corrections appearing in the brackets in (3.4d). The corrections outside the brackets are retained. These new equations can then be explicitly solved for the corrections so that iteration is unnecessary. This second procedure is a little less accurate in determining the correction but eliminates one iteration, though at the expense of introducing more complicated equations. This may be a more efficient procedure, but it is very difficult to determine definitely short of programming and comparing the two schemes.

Following Fowell, we obtain a first estimate of the desired root of (3.4a) with the following reasoning. Consider all four hypercones to be identical and defined by properties which are averages of the properties of the actual hypercones.

From a point an equal distance from the four intermediate base points lying in the initial hyperplane, project a line in the direction of the average generalized velocity, $\vec{V}_{ave}$.
The first estimate of the location of the new field point lies out along this line in the positive time direction at a distance equal to d cot μ where μ is an average Mach angle for the cones and d is the distance from one intermediate base point to the point located an equal distance from the four intermediate base points. This reasoning results in the following equations for the first estimates of the coordinates of the new field point.

$$\begin{aligned} x_5^{(0)} &= \frac{u_{ave}}{a_{ave}} d + x_c \\ y_5^{(0)} &= \frac{v_{ave}}{a_{ave}} d + y_c \\ z_5^{(0)} &= \frac{w_{ave}}{a_{ave}} d + z_c \\ t_5'^{(0)} &= \frac{U}{a_{ave}} d + t_c' \end{aligned} \qquad (3.4f)$$

where, for example,

$$u_{ave} = \frac{1}{4}(u_1 + u_2 + u_3 + u_4)$$

and the numbered subscripts indicate each of the intermediate base points. Also

$$d = \left[(x_c - x_1)^2 + (y_c - y_1)^2 + (z_c - z_1)^2 + (t_c' - t_1')^2\right]^{\frac{1}{2}}$$

the c subscripts indicate the coordinates of the point lying in the initial hyperplane an equal distance from the intermediate base points. These are determined from the following equations

$$(x_c - x_1)^2 + (y_c - y_1)^2 + (z_c - z_1)^2 + (t'_c - t'_1)^2$$
$$= (x_c - x_2)^2 + (y_c - y_2)^2 + (z_c - z_2)^2 + (t'_c - t'_2)^2$$
$$= (x_c - x_3)^2 + (y_c - y_3)^2 + (z_c - z_3)^2 + (t'_c - t'_3)^2$$
$$= (x_c - x_4)^2 + (y_c - y_4)^2 + (y_c - y_4)^2 + (t'_c - t'_4)^2$$
$$A x_c + B y_c + C z_c + D t'_c = 1$$

where the last equation is the equation of the initial hyperplane containing the four intermediate base points.

Thus, with the first estimate from (3.4f) and the double iteration process described by equations (3.4c) and (3.4d), the coordinates of the new field point can be obtained.

Step 3

The next step is to determine the flow properties at the new field point. The compatibility equations (2.15), (2.16), and (2.18c), in finite difference form, are used to determine the flow properties. In order to utilize these equations, the β-coordinates and ξ -coordinate must be determined. A representation of a portion of the field point network is depicted in Figure 13. The four intermediate base points are now redesignated P_1, P_2, P_3, and P_4 for simplicity, while P_5 is the new field point. P_6 is the point on the initial hyperplane from which the particle line through P_5 passes.

Briefly, four flow properties at P_5 are determined by using equation (2.15) written in finite difference form along the four lines from the base points to the new field point.

These four equations can be solved for u_5, v_5, w_5 and p_5. The velocity components give the direction cosines of the generalized velocity $\vec{V}$ which is tangent to the particle line. A line with direction cosines equal to the direction cosines of $\vec{V}$ is then projected back until it intersects the initial hypersurface. This intersection point is P_6. The compatibility equations (2.16) and (2.18c) along the line P_6P_5 in finite difference form determine the remaining flow properties at P_5.

The transformation needed to explicitly write the compatibility equation (2.15) has only one restriction put upon it by the partial differential equations and it is that the characteristic equation (2.6) must be satisfied. It is probably best to take the transformation to be linear and orthogonal. This does not mean that the simplest form of the required transformation is given by the orghogonal transformation. Quite to the contrary, nonorthogonal transformations could be simpler, but they are not easily determined analytically or numerically. For example, Thornhill[16] presents a nonorthogonal transformation for two-dimensional unsteady flow which is quite simply expressed. This transformation was obtained by considering the geometry of the characteristics surfaces in the three-space. Unfortunately, a direct application of the same procedure to the four-space is not possible because of the added geometrical complications of the additional coordinate.

We proceed to develop the linear orthogonal transformation which can be written in matrix notation as

$$\begin{Bmatrix} x \\ y \\ z \\ t' \end{Bmatrix} = [a_{ij}] \begin{Bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{Bmatrix} \tag{3.5}$$

The orthogonality property gives

$$[a_{ij}]^{-1} = [a_{ij}]^T = [a_{ji}] \tag{3.6}$$

The β_1-coordinate has been taken normal to the Mach hyperconoid and, hence, is normal to the line P_iP_5 at the point P_i (i = 1, 2, 3, 4). The β_2-coordinate is aligned in the direction along P_iP_5. From the work of Thornhill[16] and Bruhn and Haack[27], it is determined that $\nabla\beta_1$ and $\nabla\beta_2$ should have the following forms

$$\nabla\beta_1 = \cos\phi\,\vec{i} + \sin\phi\cos\psi\,\vec{j} + \sin\phi\sin\psi\,\vec{k} - \left(\frac{u}{U}\cos\phi + \frac{v}{U}\sin\phi\cos\psi + \frac{w}{U}\sin\phi\sin\psi + \frac{a}{U}\right)\vec{l} \tag{3.7a}$$

$$\nabla\beta_2 = \left(\frac{u}{a} + \cos\phi\right)\vec{i} + \left(\frac{v}{a} + \sin\phi\cos\psi\right)\vec{j} + \left(\frac{w}{a} + \sin\phi\sin\psi\right)\vec{k} + \frac{U}{a}\,\vec{l} \tag{3.7b}$$

where ϕ and ψ are parameters, and it is easily verified by substitution that $\nabla\beta_1$ satisfies the characteristic equation (2.6). Also, it is easily shown that

$$\nabla\beta_1 \cdot \nabla\beta_2 = 0$$

Utilizing (3.7a-b), the complete transformation is determined in the following manner. Equation (3.5) is rewritten in the following form

$$\begin{Bmatrix} x \\ y \\ z \\ t' \end{Bmatrix} = \begin{bmatrix} K_1A & K_2E & K_3L & K_4Q \\ K_1B & K_2F & K_3M & K_4R \\ K_1C & K_2G & K_3N & K_4S \\ K_1D & K_2H & K_3P & K_4T \end{bmatrix} \begin{Bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{Bmatrix} \tag{3.8a}$$

where

$$1/K_1^2 = A^2 + B^2 + C^2 + D^2 \tag{3.8b}$$

$$1/K_2^2 = E^2 + F^2 + G^2 + H^2 \tag{3.8c}$$

$$1/K_3^2 = L^2 + M^2 + N^2 + P^2 \tag{3.8d}$$

$$1/K_4^2 = Q^2 + R^2 + S^2 + T^2 \tag{3.8e}$$

There are still three degrees of freedom left in orienting the β-coordinates, so three terms can be chosen arbitrarily. The choices made here are

$$P = S = 1 \qquad T = 0 \tag{3.9}$$

The β_2-coordinate is to be aligned with P_1P_5 so equating the β_2 terms to the direction cosines of P_1P_5 gives

$$K_2E = (x_5 - x_i)/Z \tag{3.10a}$$

$$K_2F = (y_5 - y_i)/Z \tag{3.10b}$$

$$K_2G = (z_5 - z_i)/Z \tag{3.10c}$$

$$K_2H = (t'_5 - t'_i)/Z \tag{3.10d}$$

where

$$Z^2 = (x_5 - x_i)^2 + (y_5 - y_i)^2 + (z_5 - z_i)^2 + (t'_5 - t'_i)^2 \tag{3.10e}$$

Now using equations (3.7), (3.9), (3.10) and the orthogonality conditions, the terms in q_{ij} are determined with the following equations

$$H = U/a_i \tag{3.11a}$$

$$E = H(x_s - x_i)/(t'_s - t'_i) \tag{3.11b}$$

$$F = H(y_s - y_i)/(t'_s - t'_i) \tag{3.11c}$$

$$G = H(z_s - z_i)/(t'_s - t'_i) \tag{3.11d}$$

$$A = E - (u_i/a_i) \tag{3.11e}$$

$$B = F - (v_i/a_i) \tag{3.11f}$$

$$C = G - (w_i/a_i) \tag{3.11g}$$

$$D = -(Au_i + Bv_i + Cw_i + a_i)/U \tag{3.11h}$$

$$Q = (BG - CF)/(AF - BE) \tag{3.11i}$$

$$R = (CE - AG)/(AF - BE) \tag{3.11j}$$

$$L = [CF - BG + (DG - CH)R]/\mathcal{D} \tag{3.11k}$$

$$M = [AG - CE + (CH - DG)Q]/\mathcal{D} \tag{3.11l}$$

$$N = [BE - AF + (DF - BH)Q + (AH - DE)R]/\mathcal{D} \tag{3.11m}$$

where

$$\mathcal{D} = (BG - CF)Q + (CE - AG)R \tag{3.11n}$$

These equations uniquely determine the transformation at a particular intermediate base point. Using (3.6) and (3.8a), it can be seen that

$$\frac{\partial \beta_1}{\partial x} = K_1 A \qquad \frac{\partial \beta_1}{\partial y} = K_1 B \quad \ldots\ldots$$

$$\frac{\partial \beta_2}{\partial x} = K_2 E \qquad \ldots\ldots$$

so that all the partial derivatives from the transformation in (2.15) are known.

The compatibility equation (2.15) is put in finite difference form by substituting the following approximate expression for the partial derivatives along the line from P_1 to P_5.

$$\frac{\partial u}{\partial \beta_2} = (u_r - u_i)/Z \tag{3.12}$$

Similar expressions are utilized for the derivatives of v, w, and p.

A scheme must also be devised to numerically evaluate the derivatives in the β_3- and β_4-directions. This is one of the fundamental differences between problems with two independent variables where there are no partial derivatives left to be evaluated at this point, and problems with more than two independent variables. These partial derivatives can be evaluated using the following equations.

$$\frac{\partial u}{\partial \beta_m} = \frac{\partial u}{\partial x}\frac{\partial x}{\partial \beta_m} + \frac{\partial u}{\partial y}\frac{\partial y}{\partial \beta_m} + \frac{\partial u}{\partial z}\frac{\partial z}{\partial \beta_m} + \frac{\partial u}{\partial t'}\frac{dt'}{\partial \beta_m}, (m = 3,4) \tag{3.13a}$$

where from the orthogonality of the transformation

$$\frac{\partial x}{\partial \beta_m} = \frac{\partial \beta_m}{\partial x}, \quad \frac{\partial y}{\partial \beta_m} = \frac{\partial \beta_m}{\partial y}, \quad \frac{\partial z}{\partial \beta_m} = \frac{\partial \beta_m}{\partial z}, \quad \frac{\partial t'}{\partial \beta_m} = \frac{\partial \beta_m}{\partial t'} \qquad (3.13b)$$

and the other partial derivatives can be obtained from equations of the form

$$(u_s - u_i) = \frac{\partial u}{\partial x}(x_s - x_i) + \frac{\partial u}{\partial y}(y_s - y_i) + \frac{\partial u}{\partial z}(z_s - z_i) + \frac{\partial u}{\partial t'}(t'_s - t'_i) \qquad (i=1,..,4) \qquad (3.13c)$$

As a first estimate in the iteration procedure described in Step 4 below, u_5 can be taken as

$$u_5^{(o)} = (u_1 + u_2 + u_3 + u_4)/4 \qquad (3.13d)$$

In a similar manner, the derivatives of v, w, and p can be determined.

It should be pointed out that in writing equation (3.13c) it is assumed that the partial derivatives of the form $\partial u/\partial x$, $\partial u/\partial y$, $\partial u/\partial z$, and $\partial u/\partial t'$ can be approximated as constants over the network. The derivatives $\partial u/\partial \beta_m$ are not considered constant over the network in (3.13a), however, because the transformation and, hence, the derivatives given by (3.13b) are calculated for each intermediate base point.

With the equations given above, all the terms in (2.15) can be evaluated. The last term which includes the chemical source function is evaluated at least initially at each of the intermediate base points where all of the flow properties are known. Thus, with i = 1, 2, 3 and 4, equation (2.15) gives four linear

algebraic equations which can be solved for the four unknowns, u_5, v_5, w_5 and p_5.

The coordinates of the base point, P_6, are determined by the intersection of a line through P_5 with direction cosines equal to those of $\vec{V}$ and the hyperplane through the four intermediate base points, P_1, P_2, P_3 and P_4. The equations which give these coordinates, are the equation of the hyperplane

$$Ax + By + Cz + Dt' = 1 \qquad (3.14a)$$

and the equations of the line

$$\frac{x - x_5}{u_5} = \frac{y - y_5}{v_5} = \frac{z - z_5}{w_5} = \frac{t' - t'_5}{U} \qquad (3.14b\text{-}d)$$

This set of linear equations is easily solved for the coordinates of P_6. The flow field properties at P_6 can be determined by linear interpolation between the four intermediate base points in the initial hyperplane. First, transform coordinates to the x^*, y^*, z^*, t^* coordinates where the t^* coordinates are all equal to the same constant. Then, a general flow property can be written in the form

$$f = f(x^*, y^*, z^*) = F_1 x^* + F_2 y^* + F_3 z^* + F_4 \qquad (3.15)$$

and the F_i coefficients can be determined by substituting the coordinates and flow properties of the four intermediate base

points. Then the flow properties at P_6 are easily determined by substituting its coordinates.

When the flow properties at P_6 are known, the compatibility equations along the particle line, (2.16) and (2.18c), are used to determine the remaining flow properties at P_5. For this purpose, these equations might be put in finite difference form by substituting

$$\frac{dc_n}{d\xi} = (c_{n_5} - c_{n_6})/X \qquad (n = 1, \ldots, N) \qquad (3.16a)$$

$$\frac{ds}{d\xi} = (s_5 - s_6)/X \qquad (3.16b)$$

where

$$X^2 = (x_5 - x_6)^2 + (y_5 - y_6)^2 + (z_5 - z_6)^2 + (t'_5 - t'_6)^2 \qquad (3.16c)$$

(2.16) and (2.18c) can then be solved for c_{n5} and s_5.

It should be noted that in introducing the approximations of equations (3.16), we assume that c_n and s do not vary more rapidly from the initial hyperplane to the new point than u, v, w, and p. In many nonequilibrium flows, c_n and s do vary rapidly along the particle line so that in these cases, it may be advisable to integrate equations (2.16) and (2.18c) in several steps as proposed by Ferri et.al[43] and utilized in the calculations of Wood et.al.[36] . The pressure can be assumed to vary linearly along the particle line and the one-dimensional or streamtube approximations allow a Runge-Kutta technique, for example, to be used to integrate the equations from P_6 to P_5.

Sedney and Gerber[34] have also noted that utilizing entropy in nonequilibrium calculations can lead to large computation errors so that it might be advisable in some cases to use another thermodynamic variable, such as temperature, in place of the more customary quantity, entropy. A note of warning might be inserted here to indicate that the numerical integration of the reaction rate equations, can be difficult especially near equilibrium conditions. The numerical integration of such "stiff" equations is just beginning to be handled adequately at the present time.

Step 4

The final step is to set up an iteration process to increase the accuracy of the integration procedure. This is done by repeating steps 2 and 3, but using average values of the flow field properties. In equation (2.15), for example, the flow properties u_i, v_i, w_i, p_i and a_i are replaced by average values which have the form

$$u_{i,s}^{(n+1)} = \left(u_i + u_s^{(n)}\right)/2 \qquad (i = 1, \ldots, 9) \tag{3.17}$$

In equations (2.16) and (2.18c), and hence, also in (3.14) the following form of average values is used.

$$u_{s,6}^{(n+1)} = \left(u_s^{(n+1)} + u_6^{(n)}\right)/2 \tag{3.18}$$

In the iteration process, average values of the flow properties are used everywhere except in the interpolation procedure on the

initial hypersurface and for the initial conditions in the integration of the compatibility equations. The iteration process is continued until

$$\begin{aligned} u_s^{(n)} &= u_s^{(n-1)} \\ p_s^{(n)} &= p_s^{(n-1)} \\ &\vdots \qquad\quad \vdots \end{aligned}$$

to within prescribed limits or until they are equal to the ability of the computer to determine them utilizing a finite number of digits.

The iteration process described above, is similar to the modified Euler's method which is sometimes referred to as Heun's first method (see Lapidus[42], p. 88) for ordinary differential equations. When applied to ordinary differential equations, the truncation error is third-order in the step size, so that the process includes terms of second order in the step size. The process as applied here to the partial differential equations, does omit some terms which are second order in the step size through the assumption that the partial derivatives in (3.13c), are constant over the network. It can only be hoped that these neglected terms do not appreciably increase the order of the truncation error because it appears that there is no way to include the neglected second-order terms without changing the network by adding additional base points, as Butler[28] has done.

This completes the description of the fourth step and also the entire field point procedure. Next, the procedure to calculate a new point on a body hypersurface is considered.

3.3 The Body Point Procedure

The same four steps required for the field point procedure are also required for the body point procedure. The body is assumed to be specified by an equation of the form

$$\mathcal{B}(x, y, z, t') = 0 \tag{3.19}$$

With this formulation, rather general body shapes and motions can be considered. However, it is assumed here that the body hypersurface has no corners (i.e., no discontinuities in first order derivatives). Special techniques are required for handling the expansion fans and secondary shock waves which can arise because of body corners.

Three base points in the initial hypersurface adjacent to the body hypersurface, and three points which lie both in the body and the initial hypersurfaces are used to determine a new point on the body hypersurface in the following steps.

Step 1

As in the first step in the field point procedure, intermediate base points must be properly located, and the flow properties at them determined by interpolation. Three intermediate base points must be located such that the domain of dependence of the difference scheme contains the domain of dependence of the partial differential equations to ensure numerical stability. This was done by inscribing a sphere

within the tetrahedron in the field point procedure. A similar procedure could be proposed here, but it was found in doing two-dimensional unsteady flows, that some experimentation was necessary in order to properly locate the intermediate base points, so that a stable solution was obtained. Thus, it is probably better to refer to the details of the two-dimensional unsteady procedure given in Chapter 4, Section 4.3 in the description of the PATCH subroutine, and to state that some experimentation might be necessary in locating the intermediate base points.

Step 2

The coordinates of the new body point are determined by simultaneously solving the equation of the body hypersurface (3.19) together with the equations for three Mach hypercones through the three intermediate base points determined in Step 1. The equations can be solved by using the third-order Richmond iteration procedure presented in Step 2 of the field point procedure.

A representation of a portion of the body point network is shown in Figure 14. P_1, P_2 and P_3 are the intermediate base points, and P_6, P_7 and P_8 are the base points in the body hypersurface. The new body point is located at P_4, while P_5 is the point of intersection of the particle line from P_4, projected back to the plane determined by P_6, P_7 and P_8.

Step 3

The flow properties at P_4 are determined in a manner similar to Step 3 in the field point procedure. Three compatibility equations can be obtained in the form of (2.15), corresponding to the three intermediate base points. The same coordinate transformation as used for the field point can be used here. The equation for the boundary condition

$$\frac{\partial B}{\partial x} u + \frac{\partial B}{\partial y} v + \frac{\partial B}{\partial z} w + \frac{\partial B}{\partial t'} U = 0 \qquad (3.20)$$

replaces the fourth compatibility equation used in the field point procedure. One slight change is necessary to evaluate the derivatives in the β_3- and β_4-directions in (2.15) because only three and not four intermediate base points are available. One of the base points on the body hypersurface must be used as the fourth point. From the three compatability equations and the boundary condition, the flow properties u_4, v_4, w_4 and p_4 are determined.

Next, the particle line on the body hypersurface must be projected back to obtain its intersection with the initial hypersurface. The three body base points, P_6, P_7 and P_8 in Figure 14, determine a plane (not a hyperplane) which is expressed in the following form.

$$Ax + Cy + Dz = 1 \qquad (3.21a)$$

$$Ey + Fz + Gt' = 1 \qquad (3.21b)$$

The six coefficients can be determined by substituting the coordinates of the three points into the equations, and solving the resulting set of six equations. This plane is an approximation to the surface which is the intersection of the initial hypersurface and the body hypersurface. The point P_5, which is the intersection of the particle line with the plane (3.21), is determined by considering a second plane containing P_4, the generalized velocity $\vec{V}$ and the vector normal to the body hypersurface at P_4, which is given by

$$\vec{N}_B = \frac{\partial B}{\partial x}\vec{i} + \frac{\partial B}{\partial y}\vec{j} + \frac{\partial B}{\partial z}\vec{k} + \frac{\partial B}{\partial t'}\vec{l} \qquad (3.22)$$

This plane can also be expressed in the form

$$Hx + Iy + Jz = 1 \qquad (3.23a)$$

$$Ky + Lz : Nt' = 1 \qquad (3.23b)$$

The coordinates of P_5 are obtained by solving equations (3.21) and (3.23). The intersection of the two planes results in a unique point, because the plane (3.23) contains $\vec{N}_B$, which is normal to the body hypersurface. Hence, the intersection of (3.23) with the body hypersurface is the particle line, and because (3.21) also lies within the body hypersurface, the intersection is a single point.

After the particle line base point P_5 is determined, the flow properties at it are determined by interpolation in the initial hypersurface. The compatibility equations (2.16) and (2.18c) along the particle line are used to determine the

remaining flow properties at the new body point in the same manner as in the field point procedure.

Step 4

Steps 2 and 3 above are iterated to convergence in exactly the same manner as in the field point procedure.

3.4 The Shock Point Procedure

The shock point procedure is the most complicated of the three procedures considered here. Briefly, the shock point procedure can be summarized as follows. Given a point which lies both on the shock and initial hypersurfaces, together with the complete flow field incident on the shock, the location of a new point on the shock hypersurface must be found. The flow properties behind the shock at the new point and the orientation of the shock hypersurface through the point must also be determined. As in the body point procedure, three base points in the initial hypersurface adjacent to the shock hypersurface are assumed given. Actually only one shock base point which lies both on the shock and initial hypersurfaces is required to determine the new shock point, but an additional two shock base points will be required in Step 1 to determine intermediate base points as was necessary for the body point.

The shock point procedure has the same four steps as the field and body procedures: (1) locate and determine the flow properties at the intermediate base points, (2) locate the

position of the new point, (3) calculate the flow properties at the point, and (4) iterate the solution to second order in the step size.

Step 1

This step is almost exactly the same as Step 1 of the body point procedure. Also, as in the body point solution, a detailed description of a process to locate the intermediate base points could be given here, but because some experimentation will probably be necessary to assure numerical stability, the reader is referred to procedures used for two-dimensional unsteady flow in Chapter 4, which probably can be simply generalized to the four variable case.

Step 2

A representation of a portion of the shock point network is shown in Figure 15. P_1, P_2 and P_3 are the three intermediate base points which were determined in Step 1. P_4 is the base point which lies in both the shock and initial hypersurfaces. The new shock point that is to be determined is labeled P_5.

The first estimate for the location of P_5 is obtained by determining the intersection of the shock hypersurface and the three Mach hypercones from the intermediate base points. The shock hypersurface is initially approximated by the hyperplane normal to $\vec{N}_{s4}$ and passing through P_4. $\vec{N}_{s4}$ is the unit vector

shock hypersurface at P_4.

$$\vec{N}_{s_4} = N_{s_{x_4}} \vec{i} + N_{s_{y_4}} \vec{j} + N_{s_{z_4}} \vec{k} + N_{s_{t'_4}} \vec{l} \qquad (3.24)$$

The shock hyperplane then has the form

$$N_{s_{x_4}}(x - x_4) + N_{s_{y_4}}(y - y_4) + N_{s_{z_4}}(z - z_4) + N_{s_{t'_4}}(t' - t'_4) = 0 \qquad (3.25)$$

The coordinates of P_5 can be obtained by using the third order Richmond iteration scheme as in the field and body point procedures.

Step 3

The procedure for determining the flow properties behind the shock hypersurface at the new shock point is more involved than the corresponding procedures for the field and body points. First, notice that the shock hypersurface at P_5 has been approximated by the hyperplane, (3.25), so that in the first step of the interation $\vec{N}_{s5} = \vec{N}_{s4}$.

The flow incident on the shock wave at P_5 is given, so u_{5+}, v_{5+}, w_{5+}, p_{5+}, ρ_{5+} and c_{n5+} are known or can be determined by interpolation. The plus-sign subscript indicates a flow property in the incident stream while a minus-sign subscript indicates flow properties just behind the shock hypersurface. The shock wave equations (Rankine-Hugoniot equations) are used to determine the properties behind the shock.

Introducing the following notation

$$q_{N_t} = \left(u_t N_{s_x} + v_t N_{s_y} + w_t N_{s_z} + U N_{s_t}\right) \tag{3.26a}$$

where

$$\Phi^2 = \left(N_{s_x}^2 + N_{s_y}^2 + N_{s_z}^2 + N_{s_t}^2\right) / \left(N_{s_x}^2 + N_{s_y}^2 + N_{s_z}^2\right) \tag{3.26b}$$

and

$$\vec{V}_t = q_{N_t}\left(N_{s_x}\vec{i} + N_{s_y}\vec{j} + N_{s_z}\vec{k}\right)\Phi + \vec{V}_{T_t} \tag{3.26c}$$

the equations for a moving shock wave can be written in the form

$$\rho_+ q_{N_+} = \rho_- q_{N_-} \tag{3.26d}$$

$$p_+ + \rho_+ q_{N_+}^2 = p_- + \rho_- q_{N_-}^2 \tag{3.26e}$$

$$h_+ + \tfrac{1}{2} q_{N_+}^2 = h_- + \tfrac{1}{2} q_{N_-}^2 \tag{3.26f}$$

$$\vec{V}_{T_+} = \vec{V}_{T_-} \tag{3.26g}$$

$$c_{n_+} = c_{n_-} \qquad (n = 1, \ldots, N) \tag{3.26h}$$

The state equation is also needed.

$$h = h\left(p, \rho, c_1, \ldots, c_N\right) \tag{3.26i}$$

These equations cannot be solved explicitly for the properties behind the shock in terms of the properties in front of the shock, because the state equation cannot always be written explicitly in the form of an equation. The shock wave equations

must be solved numerically for a general problem, such as the flow of a real gas (e.g., for air, see Moeckel[44]).

After q_{N-} has been determined using (3.26), the velocity components are given by the following equations.

$$u_- = u_+ + N_{s_x} \Phi (q_{N-} - q_{N+}) \tag{3.27a}$$

$$v_- = v_+ + N_{s_y} \Phi (q_{N-} - q_{N+}) \tag{3.27b}$$

$$w_- = w_+ + N_{s_z} \Phi (q_{N-} - q_{N+}) \tag{3.27c}$$

Thus, all the flow properties at P_5 behind the shock hypersurfaces can be determined.

Step 4

Equations (3.26) and (3.27) determine an estimate of the flow properties at P_5 behind the shock wave, but there are three compatibility equations which hold along the lines P_1P_5, P_2P_5 and P_3P_5. In general, the properties determined above will not satisfy these compabflity equations. However, a new estimate for the shock hypersurface normal vector, $\vec{N}_{s5}$ can be made such that the conditions behind the shock tend toward the satisfaction of the compatibility equations.

The new estimate of $\vec{N}_{s5}$ is obtained as follows. The flow properties which appear in the compatibility equation (2.15) are expanded in Taylor series,which is truncated after the first order derivatives

$$u_5^{(n+1)} = u_5^{(n)} + \left(\frac{\partial u_5}{\partial N_{sx_5}}\right)^{(n)} \Delta N_{sx_5}^{(n)} + \left(\frac{\partial u_5}{\partial N_{sy_5}}\right)^{(n)} \Delta N_{sy_5}^{(n)} + \left(\frac{\partial u_5}{\partial N_{sz_5}}\right)^{(n)} \Delta N_{sz_5}^{(n)} \qquad (3.28)$$

where minus-sign subscripts have been omitted. Similar equations can be written for $v_5^{(n+1)}$, $w_5^{(n+1)}$ and $p_5^{(n+1)}$. The new estimate of $\vec{N}_{s5}$ is given by

$$\vec{N}_{s_5}^{(n+1)} = \left(N_{sx_5}^{(n)} + \Delta N_{sx_5}^{(n)}\right)\vec{i} + \left(N_{sy_5}^{(n)} + \Delta N_{sy_5}^{(n)}\right)\vec{j} + \left(N_{sz_5}^{(n)} + \Delta N_{sz_5}^{(n)}\right)\vec{k} + \left(N_{st_5}^{(n)} + \Delta N_{st_5}^{(n)}\right)\vec{l} \qquad (3.29)$$

The three correction terms $\Delta N_{sx5}^{(n)}$, $\Delta N_{sy5}^{(n)}$ and $\Delta N_{sz5}^{(n)}$ are determined by substituting (3.28) and its counterparts for v, w, and p into the three compatibility equations, (2.15), corresponding to P_1, P_2 and P_3. The partial derivatives of u, v, w, and p with respect to N_{sx}, N_{sy} and N_{sz} must be determined before the correction terms can be obtained. These derivatives may be obtained from equations (3.26) and (3.27), but they cannot be determined analytically unless the state equation can be written explicitly. They can be determined numerically, and perhaps, this is the simplest way even when they can be determined analytically. For example, a derivative could be calculated using the following approximation.

$$\frac{\partial p}{\partial N_{sx}} = \frac{p(\vec{N}_s + \delta N_{sx}\vec{i}) - p(\vec{N}_s)}{\delta N_{sx}}$$

where δN_{sx} is a small change in N_{sx}. After the three corrections are determined, the fourth correction is obtained from

$$\left(N_{s_{x_5}} + \Delta N_{s_{x_5}}\right)^2 + \left(N_{s_{y_5}} + \Delta N_{s_{y_5}}\right)^2 \tag{3.30}$$

$$+\left(N_{s_{z_5}} + \Delta N_{s_{z_5}}\right)^2 + \left(N_{s_{t'_5}} + \Delta N_{s_{t'_5}}\right)^2 = 1$$

This procedure is repeated in an iteration process as in the field and body point procedures. In Step 1, $\vec{N}_{s4}$ is replaced by an average value given by

$$\vec{N}_{s_{4,5}}^{(n+1)} = \frac{1}{2}\left[\left(N_{s_{x_4}} + N_{s_{x_5}}^{(n+1)}\right)\vec{i} + \left(N_{s_{y_4}} + N_{s_{y_5}}^{(n+1)}\right)\vec{j}\right.$$

$$\left. + \left(N_{s_{z_4}} + N_{s_{z_5}}^{(n+1)}\right)\vec{k} + \left(N_{s_{t'_4}} + N_{s_{t'_5}}^{(n+1)}\right)\vec{\ell}\right] \tag{3.31}$$

where $\vec{N}_{s5}^{(n+1)}$ is given by (3.29). In Step 2, note that $\vec{N}_5^{(n+1)}$ is not equal to $\vec{N}_{s4,5}^{(n+1)}$ but is given by (3.29). Average values are used in the hypercone and compatibility equations just as in the field and body point procedures.

3.5 The Initial Value Hypersurface

The Cauchy problem being considered here requires that the flow properties be completely specified on an initial hypersurface. This means that a three-dimensional flow field must be specified before the complete four-dimensional flow field can be solved. By no means, is this a trivial problem. Of course, the type of initial value hypersurface required for a given problem will depend on the specific problem. A few of the properties which seem unique to the general method of characteristics and which might be used to determine the initial hypersurface are mentioned in the following discussion.

In most practical problems of unsteady three-dimensional flow, the initial hypersurface is the initial steady three-dimensional flow. Thus, the steady flow is required before the unsteady flow can be determined. This brings to mind, specializing the general method of characteristics to solve the three-dimensional steady flow. This is discussed in the next section. Only steady supersonic flow can be solved by the method of characteristics, because the equations of steady subsonic flow are not hyperbolic, but this is one way in which the initial hypersurface might be determined for supersonic flow.

Certain steady flows might also be solved in the following manner. Consider the hypersonic flow about a blunt nosed body as depicted in Figure 16. The flow in the initial hypersurface, $t' = 0$, may be only an estimate of the exact flow. The flow can then be calculated in the t'-direction with the body surface unchanged with time. The initial estimate can be considered physically as the result of some body motion for $t' < 0$. After a certain amount of time has passed, the transient decays and steady flow is attained at $t' = t'_s$. For $t' > t'_s$, the flow field does not change with time. In this way, the steady flow can be calculated starting from only an estimate of the flow field. Note that the initial estimate must not violate the equations of change of the gas. Hence, the initial estimate is not arbitrary, but must be physically realizable. Obtaining such an estimate for nonequilibrium flow may not be simple, but perhaps rather crude initial estimates would still yield the physically correct results. This could be ascertained by solving test cases.

The difficulty in obtaining an initial estimate which is compatible with the equations of change could be avoided by using the following technique. Start with an initial steady flow which is known exactly. For example, the hypersonic flow about a spherically capped cylinder might be taken as the known steady flow as shown in Figure 17. The spherical cap can then be distorted into some other shape, perhaps ellipsoidal, and after the transient decays, the steady flow about the distorted body is obtained. In the same way, the known initial steady flow might be taken as the supersonic flow about a cone at zero angle of attack. The cone can then be rotated to a finite angle of attack, and after the transient decays, the steady flow about a cone at an arbitrary angle of attack is obtained. In this way, the steady supersonic flow about some rather arbitrary bodies can be calculated.

It should be mentioned that the general method of characteristics for unsteady flow applies to subsonic as well as supersonic flows. Given the initial steady subsonic flow, the flow-field history can be computed within the domain of determinacy of the initial data. The domain of determinacy is limited in the t'-direction by the amount of initial data given in the initial hypersurface, but the calculation can be carried as far in the t'-direction as desired, merely by enlarging the region of the initial data. For practical reasons discussed in Chapter 5, the method of characteristics for unsteady flows will probably never be used for subsonic flows unless the flow is desired for only a very short period of time.

3.6 Simplifying the General Procedure

It will not always be necessary to consider the three-dimensional unsteady flow of a nonequilibrium gas. The method of characteristics can also be applied to simpler problems. Means for specializing the general procedure can be very useful and should be kept in mind as the general procedure is developed. Specifically, three simplifications are considered here. They are listed below, together with a brief discussion of how the simplification is carried out.

Nonequilibrium flow simplified to equilibrium flow. In this case, the simplification is accomplished by eliminating the following variables and operators : σ_n, Dc_n/Dt' and $\partial/\partial c_n$. Also, the equilibrium speed of sound, a_e, should be substituted for the frozen speed of sound. This should be a rather simple process.

Three-dimensional unsteady flow simplified to two-dimensional unsteady flow. Here, one of the space coordinates is to be dropped so, for instance, the following terms are set equal to zero: z, $\partial/\partial z$, w, β_4 and $\partial/\partial\beta_4$. In the coordinate transformation, ψ is also set equal to zero. This too seems to be a rather simple process.

Three-dimensional unsteady flow simplified to three-dimensional steady flow. In this case, the specialization is not easily accomplished. First, time must be eliminated from the problem by setting t' and $\partial/\partial t'$ equal to zero. The

characteristic equation, (2.6) changes to

$$\left(u\frac{\partial\beta_1}{\partial x}+v\frac{\partial\beta_1}{\partial y}+w\frac{\partial\beta_1}{\partial z}\right)^2-a^2\left[\left(\frac{\partial\beta_1}{\partial x}\right)^2+\left(\frac{\partial\beta_1}{\partial y}\right)^2+\left(\frac{\partial\beta_1}{\partial z}\right)^2\right] \tag{3.32a}$$

so that

$$u^2+v^2+w^2 > a^2 \tag{3.32b}$$

for the problem to remain hyperbolic. The equation of the Mach cone changes to

$$\begin{aligned}&(v^2+w^2-a^2)(x-x_i)^2+(u^2+w^2-a^2)(y-y_i)^2\\&+(u^2+v^2-a^2)(z-z_i)^2-2uv(x-x_i)(y-y_i)\\&-2uw(x-x_i)(z-z_i)-2vw(y-y_i)(z-z_i)=0\end{aligned} \tag{3.32c}$$

Note that (3.32c) is not obtained from (2.9) merely by setting t' equal to zero. In order for the Mach hypercone to be written in a general form so that it can be easily simplified, the characteristic equation, (2.6), must be written in the form

$$\sum_{n=1}^{4}\sum_{m=1}^{4} b_{mn}\frac{\partial\beta_1}{\partial\alpha_m}\frac{\partial\beta_1}{\partial\alpha_n}=0 \tag{3.32d}$$

where

$$(\alpha_1,\alpha_2,\alpha_3,\alpha_4)=(x,y,z,t')$$

and then the Mach hypercone can be written in the form

$$\sum_{n=1}^{4}\sum_{m=1}^{4} A_{mn}(\alpha_m-\alpha_{m_i})(\alpha_n-\alpha_{n_i})=0 \tag{3.32e}$$

where A_{mn} is the matrix made up of the cofactors of the determinant of b_{mn}. When t' is set equal to zero in (3.32d), equation

(3.32e) will give the correct equation for the Mach cone.

Another complication in specializing to steady flow is that the coordinate transformation, (3.8a), cannot be easily simplified to steady flow. Hence, a different transformation such as the one used by Ferri[13] and Fowell[23] is required for the steady flow. Because of the above difficulties, it might be easier to develop a procedure specifically for three-dimensional steady flow, such as the work reported by Fowell, than to try to specialize three-dimensional unsteady flow. This speculation can be verified when the general procedure is programmed for the digital computer.

CHAPTER 4

THE PROGRAMMING OF THE PROCEDURE

In this chapter, the details of the programming of the method of characteristics are presented. This is the portion of this study which pertains to the second part of the object stated initially, that is, the proving (or disproving) of the feasibility of the method by programming and utilizing the method to solve practical flow fields. This is by no means a minor portion of the total project.

4.1 Practical Considerations

Many factors were considered in deciding exactly what programs could be feasibly attempted with presently available high speed electronic digital computers. First, the limitations of the IBM 7094 to be utilized for the calculations had to be considered. This computer has approximately 32,000 words of magnetic core storage available. About half of this storage should be set aside for the programs, which leaves 17,000 words for the storage of data on the initial hypersurface. It has been found that the new data can be written over the initial data as the calculation progresses if a few temporary storage locations are used. In this way, no more than one hypersurface of data needs to be stored at one time. A few simple calculations can check the adequacy of this storage for various problems. For three-dimensional unsteady nonequilibrium flow, (9 + N) words of data (the values of x, y, z, t', u, v, w, p,

s, and c_n) must be stored for each point. If N is eight, for example, data could be stored for 1000 points. In the initial hypersurface, this would give 10 points in a typical linear dimension, which hardly seems adequate to describe a practical flow field. If equilibrium flow is considered, 1900 points could be stored for the initial hypersurface, but still this would give only 12 points in a typical linear dimension. Finally, consider a two-dimensional unsteady equilibrium flow which requires seven words of data at a point. This would allow 2500 points in the initial surface, or 50 points in a typical linear dimension. This is a more reasonable number of points to describe a flow field.

Another machine limitation is the operating speed of the computer. Even though calculations can now be carried out in a time on the order of microseconds, programmers have been able to demand more and more of computers,as in the problem considered here. Computation time must be considered, because this is what determines the financial cost of carrying out the calculation. A quantitative estimate of operating time is not possible before the programs are written, but it is possible to draw some qualitative conclusions. One result from two-dimensional steady flow calculations, is that nonequilibrium and real gas thermodynamic subprograms tend to require relatively larger amounts of computer operating time. The complicated iterations discussed in Chapter 3 will require large amounts of operating time, but these cannot be eliminated without reducing the calculations to rather trivial

problems. Thus, it appears advisable to consider a perfect gas as a first case to keep operating time reasonable.

The programs were written in the FORTRAN II computer language by Mark Sussman, and the author. Neither of the programmers was familiar with the FORTRAN language when the project was begun. Some, or perhaps, most of the programs if written in a more machine oriented language, such as FAP would require less storage and less operating time. However, due to the inexperience of the programmers, it was decided that the more mathematically oriented FORTRAN language could be utilized in a much shorter time period.

With the above mentioned considerations in mind, it was decided that the general problem formulated in Chapter 3 must be limited slightly, in order to write some useful programs at the present time. These limitations have been incorporated in such a way that the difficulty in removing them should not be great, or should not require any fundamental changes in the programs. First, the problem was limited to two-dimensional unsteady flow as discussed in Section 3.6. This reduces the storage required to a reasonable quantity, and is rather simply carried through. This restriction is also rather easily removed and in most equations (and therefore, also in program statements), the neglected terms which would contain the third space coordinate can be re-introduced by inspection.

To keep computing times reasonable, it was decided to consider only a perfect gas in this initial study. Equilibrium

real gas thermodynamics can be introduced merely by replacing the perfect gas thermodynamic subroutines with their real gas counterparts. The addition of nonequilibrium effects will require different chemical and thermodynamic programs, and the modification of some of the programs to include the integration of the species continuity equation along the streamline.

There is also an argument for simplifying the problem, other than because of programming restrictions. The problem should be simplified as much as possible while still retaining its major features, so that complexities from auxiliary sources such as thermodynamic subroutines do not overshadow the more important effects which are to be investigated. Thus, initially one space dimension can be dropped and a perfect gas considered. Then, after the programs for this problem are working the further complexities of the third space dimension and more complicated thermodynamics can be added.

The problem to be considered is still very challenging, because three independent variables are to be considered. Few people as yet have had success in using the method of characteristics for problems in three independent variables. Only Butler[28] and Talbot[45], utilizing Butler's formulation have considered two-dimensional unsteady flow. Butler's approach is not exactly the method of characteristics as pointed out in Section 3.1, and it is not obvious that it can be easily extended to four independent variables. The resulting programs will also allow the checking of some of the proposed calculation schemes discussed in Section 3.5.

4.2 The General Approach

Three subroutines that determine the location and flow properties at the field, body,and shock points are the heart of the programmed procedure. These are the most important subroutines and all of the other subroutines and the executive program (or main program, as it is termed in programming manuals) merely support and assist these subroutines in calculating the flow fields.

The executive program organizes the points in the initial surface and calls the three principle subroutines to calculate new points on the next surface. It then accepts the results from the subroutines and organizes the data in the next surface. These surfaces will be referred to as time surfaces, because they are usually close to being t' = constant planes. The more general a type of flow the executive program can handle; the more complicated and sophisticated it must be. In this initial study, the executive was kept as simple as possible and because of this, it is very much oriented to the particular type of flow field being solved. Hence, if a completely different type of flow is to be calculated, a new executive program would be required.

It should be pointed out that the programs were written with the three basic subroutines for field, shock, and body points being supported by all other programs, because these are the fundamental elements in the calculation, and are universal to all inviscid flow problems. Thus, they can be

utilized in all other flow calculations of this type, merely by supplying the proper supporting programs. If the calculation of a new field, shock, and body points had been combined in larger programs together with other supporting calculations, oriented toward specific flow fields and boundary conditions, they could not have been utilized as readily in different flow calculations.

The general program organization is shown in Figure 18, in the form of a calling sequence diagram. The three subroutines to calculatie field, body, and shock points have been given the mnemonic names FLDPT, BDYPT, and SHKPT. The programs can be divided into four general sections as shown in the figure. The executive is the first section. In the second section, there are four subroutines which control interpolation, the number of points in a surface, and reading or writing data on the tape. The third section has the subroutines which actually calculate new data points. In the last section, there is only one subroutine which does an auxiliary interpolation. Three FORTRAN functions are also utilized. One is named THETA, which is called by TRUNET and PATCH. The last two functions carry out the thermodynamic calculations and are called ADET and ROEDET. A more detailed discussion of each of the subroutines is given in the next section.

4.3 Detailed Discussions of the Programs

In the following discussions, only those techniques and procedures which have not been discussed in Chapter 3 will be presented. General flow diagrams and complete program listings are presented in Appendix C. These discussions are intended to point out the function of the programs, the checking procedures built into them and any unusual or special techniques which were found necessary to insure successful operation. A knowledge of the FORTRAN II programming language and the IBM 7090/94 Data Processing Systems is assumed (see References 46 and 47).

Executive Program

As mentioned in Section 4.2, the executive program is oriented toward solving a particular flow field. The executive considered here and listed in Appendix C is for the flow in the subsonic, transonic and supersonic regions between a detached shock wave and the surface of a blunt body moving at high supersonic speed. This program is an improved version of an earlier program which was used to calculate the flow over a wedge in a supersonic or hypersonic flow.

This executive reads the data for the initial surface from a tape where it had been previously written in the proper order and format. The data is stored in the common portion of the magnetic core storage in three large arrays, named FLPTS1, BDPTS1 and SKPTS1 for field, body, and shock points, respectively.

The executive merely transfers various words of data from these locations to other locations in the common storage, where the PATCH, FLDPT, BDYPT and SHKPT subroutines pick up the input data for their calculations.

Four data cards are read by the executive to specify auxiliary parameters. The first card contains numbers which specify what the program is to do on a particular run, as explained in the listing. Free stream conditions are specified on the second card and the nondimensionalizing velocity, U, together with the convergence test values are specified on the third and fourth cards.

The organization in the executive calculates data on an odd numbered surface which contains only field points. It then uses these points together with the shock and body points from the initial surface to calculate field, shock, and body points on an even numbered surface. The detailed organization of the field points is shown in Figure 19. The new shock and body points are calculated using two field points from the rows closest to the shock and body surface on the odd numbered surface. The body and shock points required as base points are taken from the even numbered surface.

After the executive has sequencially calculated the specified number of new time surfaces, it writes the final data on tape. This data is in such a form, that the executive can read it from the tape at a later time and continue the calculation.

SCAN

This is a FAP coded subroutine which merely positions the data tape so that the proper data can be read from it or written upon it. The data is written on the tape in logical records separated by end of file marks. Each logical record is the data for one initial surface. SCAN has one integer argument which specifies the logical record on the tape which is to be read or written.

ADDROW

As can be seen from Figure 19, one row of points is lost for each step taken to a new surface. It is possible that in certain flows, such as the blunt body case considered here, that these points will be spread over a larger area at each step. The density of the points will then decrease. ADDROW is used to add more rows of points between the existing boundary rows by linear interpolation. The rows must have been evenly spaced by TRUNET before ADDROW is called, because ADDROW assumes even spacing in doing its calculation,though slignt variation in spacing can be tolerated by ADDROW.

TRUNET

The location of points on a new surface is a function of the spacing of the points on the initial surface, the velocity field and the speed of sound throughout the field. Because the

last two factors cannot be controlled, the locations of the points in the new surfaces tend to drift out of their initial configuration after a few steps to new time surfaces have been made. TRUNET uses linear interpolation to "true up" the net in both time and space. It can be called periodically after a few steps to restore the net to its original configuration.

TRUNET is just a slightly different version of the executive program. It uses an organization similar to the executives which is shown in Figure 19, but it calculates new points in a slightly different sequence, such that it can interpolate between two even numbered surfaces. TRUNET first interpolates in the time direction between corresponding points on the two surfaces obtaining the data at points on a constant time plane. It then equally spaces field points on lines connecting corresponding shock and body points in the constant time plane. Flow properties at these new field points are obtained by linear interpolation between three adjacent points in the constant time plane, which were previously determined in the interpolation in the time direction. Hence, TRUNET calculates two new surfaces (an even numbered, and an odd numbered surface), but also interpolates to obtain the data on a constant time plane at equally spaced points.

TINTER

This function merely specifies the constant time at which TRUNET is to place the constant time plane. This time should be somewhere between the two even numbered surfaces that are used in the interpolation in TRUNET. The time is usually determined by calculating a typical point on the new surface being determined. If the time specified is not between the two surfaces, extrapolation rather than interpolation will be carried out in TRUNET. The specification of the time was removed from TRUNET to the small subroutine, TINTER, to facilitate easy change of the specified time.

PATCH

The PATCH subroutine accepts the base points as input, and determines the location and flow properties at the intermediate base points, so that the domain of dependence of the difference scheme contains the domain of dependence of the partial differential equations. This insures the numerical stability of calculation (see Appendix B).

PATCH has four different options for locating the intermediate base points corresponding to values of 1, 2, 3, and 4 of its first argument. Its second argument is the convergence test value used in the iteration loop. The first option locates the intermediate base points for the field point procedure, as shown in Figure 20(a), and discussed in detail in Chapter 3. The second option was an unsuccessful scheme for the body and

shock point procedures. A circle was fitted as shown in Figure 20(b), and the points of tangency between the field points and body (or shock) points were used as the location of the intermediate base points. This scheme did not completely satisfy the stability criterion and a mild instability was discovered in the results, after 10 or more time surfaces had been calculated.

The third and fourth options are successful schemes for the shock and body point procedures, respectively. They both use the pattern shown in Figure 20(c). First, an intermediate shock (or body) point is determined as an average of the shock (or body) base points. The intermediate base points are then located four tenths of the way up the line from this intermediate shock (or body) point to the field base points. The only difference between the shock and body point options is that the intermediate shock point is used in SHKPT, while the intermediate body point is not used in BDYPT.

Linear interpolation is used everywhere in PATCH. This part of determining the location of the intermediate base points was not included in the FLDPT, BDYPT and SHKPT subroutines, so that if higher order interpolation were desired at a later time, it could be easily included by merely writing a new PATCH subroutine.

PATCH has three error checking procedures in the first two options where iteration is necessary. If the iteration does not converge in 50 cycles through the loop, the calculation is

halted, control is transferred back to the calling program, and an error statement is printed off-line. The sets of linear algebraic equations are solved using the M.I.T. Computation Center library routine XSIMEQ. If over or underflow occurs while the equations are being solved or if the matrix of the equations is singular, the computation is halted and appropriate error statements are printed off-line. XSIMEQ is used in FLDPT, BDYPT and SHKPT, and the same error checking procedure is used in those subroutines.

FLDPT

The FLDPT subroutine calculates the coordinates and flow properties of a new point in the field, given the intermediate base points calculated by PATCH. It does this using three iteration loops; two for the third order Richmond scheme, to solve the Mach cone equations, and one to converge the solution to the compatibility equations to second order in the step size.

The convergence tests, as proposed in Chapter 3, have the form

$$\frac{f^{(n)} - f^{(n-1)}}{\text{suitable normalizing factor}} < \text{convergence test value} \qquad (4.1)$$

where f is a typical quantity to be converged, and the inequality must be fulfilled before the sequencing is released from the loop. Typical convergence test values range from 0.0001 to 0.000001. These tests are employed on the inner and outer loops, but it was found that the middle loop, which

tests the coordinates of the new point, sometimes could not converge them to typical convergence test values. This was because the residuals in the Mach cone equations had already been reduced to zero,to eight decimal digits which is the ultimate accuracy of the machine using single precision arithmetic. For this reason, the residuals in the equations are tested in the middle loop tests where typical convergence test values of 0.0000005 are used.

As error checks, FLDPT has the same 50 cycle limit on its looping as PATCH. The SHKPT and BDYPT subroutines also have the same limits. In each case, control is returned to the calling program, and an error statement is printed off-line together with the x, y, and t' coordinates of the point which did not converge.

SHKPT

The SHKPT subroutine calculates the coordinates and flow properties of a new point on the shock wave, together with the components of the shock wave normal vector. It has three iteration loops as in the FLDPT subroutines. The error checking procedures and convergence tests are also exactly the same as those in FLDPT.

It was found that in using equation (3.30) to calculate $\Delta N_{st'}$, even though the quantity under the radical did not converge to a negative number, it was possible for it to take on a negative value during the outer loop iteration. This, of course,

caused the machine to error stop because it was not programmed to handle the imaginary number. This trouble was eliminated by testing N_{sx} and N_{sy} and using the smaller together with N_{st}, in the expansion, (3.28). Eliminating the largest component and solving for its correction using equation (3.30), insures that the quantity under the radical will not turn slightly negative during the iteration. This procedure has been incorporated in the programs and seems to eliminate the trouble.

BDYPT

This subroutine calculates the coordinates and flow properties of a new point on the body surface. It uses the same three iteration loops, convergence tests and error checking procedures as the SHKPT and FLDPT subroutines described above.

It was found that in certain expansion regions on the body surface, the pressure could become negative in the outer loop iteration even though it finally converged to a positive quantity. This would cause an error stop in the ROEDET or ADET functions where the pressure is raised to a nonintegral power. To eliminate this trouble, the pressure is set equal to zero if it turns negative. BDYPT checks the final value of the pressure before returning control to the calling program, and prints a warning statement off-line if the final converged value of the pressure is zero.

XCHANG

All references to the function

$$B(x, y, t') = 0 \tag{4.2}$$

which describes the body surface have been removed from the BDYPT subroutine and placed in a subroutine called XCHANG. This was done to minimize the number of changes necessary to consider different motions of the same body or in certain cases, the flow about different bodies. The general form of XCHANG is given in the form of a listing in Appendix C.

XINTER

This subroutine uses linear interpolation to determine a flow property at a fourth point given the coordinates of three other points at which the flow property is known.

ROEDET

This function utilizes the perfect gas equation

$$\rho = \rho_\bullet \left(\frac{P}{P_\bullet}\right)^{\frac{1}{\gamma}} e^{-\left(\frac{S-S_0}{R}\right)\left(\frac{\gamma-1}{\gamma}\right)} \tag{4.3}$$

where

p_o = 2117 lb./ft.^{2}

ρ_o = 0.002498 slug/ft.^{3}

γ = 1.4

s_o/R = 41.11

R = 0.06886 BTU/lbm.$^\circ$R = 1724 ft.^{2}/sec^2 $^\circ$R

to compute the density given the pressure and entropy.

ADET

Given the pressure and entropy, this function determines the speed of sound using the perfect gas equation

$$a_e^2 = \gamma \frac{p_0}{\rho_0}\left[\frac{p}{p_0} e^{\left(\frac{s-s_0}{R}\right)}\right]^{\left(\frac{\gamma-1}{\gamma}\right)} \qquad (4.4)$$

where the constants are given above.

THETA

This is a very simple function which determines an angle in radians when given the opposite and adjacent sides of a right triangle.

CHAPTER 5

RESULTS AND CONCLUSIONS

5.1 The Feasibility of Calculating Practical Flow Problems

The feasibility of carrying out a practical calculation of an unsteady flow field hinges on whether the flow can be calculated in a reasonable number of time steps. An estimate of the number of time steps required can be obtained in the following manner. The total time required for a practical calculation can be expressed as

$$t_{Tot} = \frac{KL}{V_{ave}} \tag{5.1}$$

V_{ave} is an average particle velocity in the field, L is a typical body length and K is the number of particles which must pass over the body during a typical transient motion or during one cycle of an oscillation. K can also be considered the number of body lengths a particle must travel during a typical transient motion. Now the average time step is related to the space mesh size by the following well known estimate of its order of magnitude.

$$\Delta t = O\left[\Delta x / a_{ave}\right] \tag{5.2}$$

This can be easily derived by considering the geometry of the characteristics. Now an estimate of the total number of time steps is given by

$$\tau = \frac{t_{Tot}}{\Delta t} = O\left[\frac{KL\, a_{ave}}{\Delta x\, V_{ave}}\right] = O\left[\frac{K}{\varepsilon M_\infty}\,\frac{L}{\Delta x}\right] \tag{5.3}$$

where

$$\varepsilon = \frac{V_{ave}}{a_{ave} M_\infty} = \frac{M_{ave}}{M_\infty} \tag{5.4}$$

Now τ can probably vary between the extreme cases of

$L/\Delta x = 20 \qquad K = 5 \qquad \varepsilon = \frac{1}{2} \qquad M_\infty = 20$

which gives a τ of 10 to

$L/\Delta x = 50 \qquad K = 10 \qquad \varepsilon = \frac{1}{10} \qquad M_\infty = 5$

which results in a τ of 1000. Hence, it appears that higher Mach number problems which do not require too fine a mesh spacing, $\Delta x/L$, can be attempted with this method. With the present programs, which utilize linear interpolation, the upper limit on τ seems to be approximately 50 because of growing errors introduced by the interpolation. The use of higher order interpolation should increase this limit by a factor of perhaps three to four.

Another area where the solution of two-dimensional unsteady flow arises is in hypersonic small-disturbance theory[48]. In this theory the flow over a three-dimensional body is simplified to consider the two-dimensional unsteady flow in plane slabs which sweep over the body at the free stream velocity. An estimate of the number of time steps required to do this sort of problem is given by

$$\tau = \frac{t_{TOT}}{\Delta t} = O\left[\frac{L}{\Delta x}\frac{a}{V_\infty}\right] = O\left[\frac{L/\Delta x}{M_\infty}\right] \qquad (5.5)$$

Even for $L/\Delta x = 50$ and $M_\infty = 5$, only 10 time steps are necessary to do the problem. Hence, it appears that most problems utilizing hypersonic small-disturbance theory can be easily handled.

5.2 Operational Results for the Programs

The results of a typical field point calculation are presented in Table I. These results from the FLDPT subroutine

are typical in the amount of looping required, the rate of convergence and the accuracy obtained in the iterations. Similar results are obtained with the BDYPT and SHKPT subroutines. It should be noted that the inner loop usually requires three cycles of the iteration while the middle loop usually satisfies the convergence criterion the first time through the loop. The outer loop, which in general governs the amount of computation time required, usually takes four to six cycles of the iteration to converge. The results are given to eight digits in Table I because this is the number of digits used by the machine in the calculation. Only the first four or five digits are significant, however.

It can also be seen from Table I that the calculation is converging very rapidly for at the bottom of the table are presented the results of the same calculation with smaller convergence test values. These results indicate that the inner loop, for example, has actually been converged to a convergence test value one order of magnitude smaller than required in the first calculation. An additional two cycles through the outer loop satisfies a convergence test value two orders of magnitude smaller than the original value.

These results might indicate that the convergence test criteria could be eliminated and the calculation carried out with a fixed set of cycles through each loop. This, however, is not recommended without a very thorough study of the ranges of values of all input parameters. It is very possible that for

certain areas of flow fields or certain network spacings additional iteration might be necessary to obtain reasonable accuracy.

The computation time required to determine the coordinates and flow properties at a new point is very difficult to determine precisely. It depends on the convergence test values and on the variation of the flow properties at the base points (i.e. the size of the net spacing). Estimates of the time required to compute one point have been obtained in the following manner. The total time required to read the data from the tape, calculate two new time surfaces with a total of 600 to 1000 points and write the data back on tape was divided by the total number of points calculated. In general the time ranged from 0.25 to 0.45 seconds per point with an average approximately in the middle. This time, of course, includes body and shock points and also the time required to load the input data into the proper subroutines. It is, however, a conservative estimate of the time required to determine a new field point.

The executive program and subroutines listed in Appendix C together with required input-output, library, and error tracing routines take up about 20,000 words of the magnetic core storage. This does not include the common storage area where all the data is stored. This storage requirement can be reduced by improving the subroutines as discussed in Section 6.1. Several error checking and debugging routines brought in from the library could also be eliminated. It is felt that

the minimum storage required for the programs lies somewhere between 10,000 and 15,000 words, perhaps closer to the larger figure. Thus, the original estimate in Chapter 4 that about half of the 32,000 word core storage would be needed for the storage of programs seems correct.

The FLDPT, BDYPT, and SHKPT subroutines were individually checked initially using input data obtained from analytical solutions to rather simple flow fields. The output was then checked against the known analytical solutions. The one-dimensional unsteady motion of a piston in a cylinder of gas was used to check the FLDPT and BDYPT subroutines. The motion of normal and oblique plane shock waves was used to check SHKPT. These checks were not exhaustive in that they covered large ranges of the input parameters. Due to time limitations the checks were carried out only until reasonable assurance was obtained that the subroutines were working properly. The subroutines were given additional checks in calculating complete flow fields and the results of these checks are given in Section 5.4.

5.3 Difficulties Encountered and Their Solutions

In the course of programming and debugging this method of characteristics several difficulties were encountered which had to be overcome or avoided in order to carry out practical calculations. It is felt that a brief description of these difficulties might help those who attempt to utilize or extend the procedures presented here.

Probably the most fundamental difficulty encountered was the numerical instability which arose when the tetrahedral characteristic line network was used initially. This trouble was eliminated by gaining an understanding of the basic cause of the instability (see Appendix B) and then utilizing this understanding to propose a modification to the network such that stability was obtained. The modified tetrahedral characteristic line network was the solution to this difficulty and the PATCH subroutine was the programmed form of the solution.

It was pointed out in Chapter 4 that the location of a new point is a function of the spacing of the base points in the initial surface, the local flow velocity and the local speed of sound. The last two factors cannot be controlled so that, if no effort is made to control the positions of the base points, the net can become skewed. This in fact is what happened in all the flow fields calculated. After several time surfaces had been determined the last surface on which the flow was know became skewed so that it was no longer close to being a constant time plane. When the skewing got very bad the approximations which are used to determine the first estimate of the coordinates of the new point give very poor estimates. The iteration process to determine the coordinates of the points then takes a large number of cycles to converge and eventually is unable to converge at all. Hence, in order to be able to continue the calculation, the skewing must be controlled or eliminated. The only way to do this is to control the location of the points in the initial surface. This was done by using interpolation periodically to

eliminate the net skewing. The TRUNET and ADDROW subroutines described in Chapter 4 were written to perform this interpolation and to redistribute the points in the initial surface.

Some difficulty was encountered in choosing the proper form of the convergence tests in the three loops. As discussed in Chapter 4, it was found that for some of the smaller net spacings the residuals in the Mach cone equations were reduced to zero to eight decimal digits before the coordinates had converged to four or five digits. Hence, it was found necessary to utilize residual tests in the middle loops of the FLDPT, SHKPT, and BDYPT subroutines. The outer loop tests also caused some trouble in these subroutines when one of the velocity components was close to zero. This trouble was eliminated by testing either the pressure or the total velocity magnitude, whichever was greater. In this way there is almost no chance of trying to test a quantity whose magnitude is almost zero. Quantities which pass through zero are difficult to use in convergence tests because they cannot be suitably normalized.

5.4 The Results for Complete Flow Fields

Blunt Body Flows

The flow over a circular cylinder with its axis normal to a supersonic free stream has been considered as an interesting and challenging complete flow field to study. This flow is of interest in that it has a detached shock wave which bounds the field on one side and the body surface which forms the boundary on the other. The flow field is challenging because it contains both subsonic and supersonic regions and hence also has large

variations in the flow properties. Obtaining the initial steady flow, from which unsteady flows can be calculated, is by no means a simple matter. The steady flow involves the solution of a mixed elliptic-hyperbolic problem with a free boundary. Belotserkovskii[49] has solved this steady flow numerically using the method of integral relations.

Figure 21 shows the initial data net used for the blunt body flows presented here. These points and the flow properties at each of the points were taken from the tabular results presented by Belotserkovskii. The net was made more uniform by adding four rings of points by linear interpolation. The properties were given to three or four digits in the tabular presentation. Two strips were used in the method of integral relations to obtain this initial flow. A complete symmetric net was obtained by reflecting the points about the x-axis.

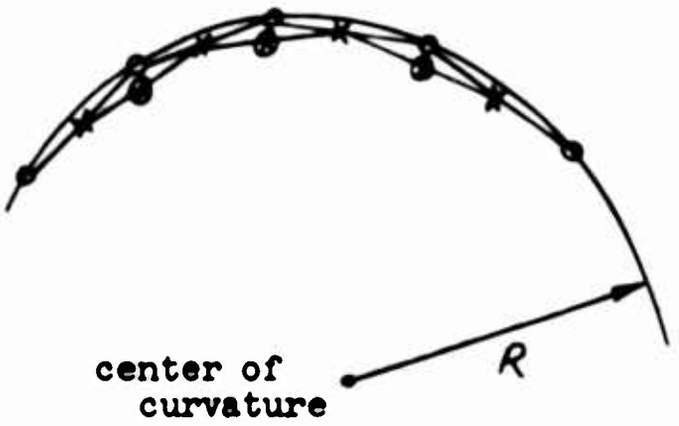
The highest free stream Mach number for which Belotserkovskii presented results was five. Unfortunately, when (5.4) is applied to this low Mach number case (with $K = 5$, $\varepsilon = 0.1$ and $\Delta x/L = 0.05$) the estimated total number of time steps required for a typical problem is 200. Obviously this is not a hypersonic flow and a complete calculation would require an excessive amount of computation time. For example, with the net of 350 points considered here, 200 time steps at one third of a second per point would require six and one half hours of computer time. Even if this large amount of computer time were available, the linear interpolation utilized in these initial programs would

introduce an increasing inaccuracy which would make the solution obtained after as many as 200 time steps only qualitatively correct. However, because of the unavailability of higher Mach number solutions of two-dimensional perfect gas blunt body flows, the $M_\infty = 5$ data of Belotserkovskii can be used to check the feasibility of carrying out practical calculations by considering at least the initial part of some unsteady motions. The results obtained starting with this initial data and considering several types of unsteady body motion are discussed in the following.

Four different blunt body flow cases were calculated for presentation here. A steady flow was calculated by holding the body surface unchanged with time. The second case warped the body symmetrically from a circular cross-section to an elliptic cross-section in a short length of time. A similar case was run with the warping done asymmetrically so that the final elliptic section was at an angle of attack of 20°. The fourth case oscillated the body vertically (in the y-direction) with a $1 - \cos \omega t$ time dependence.

The steady flow was run to ascertain how well the calculation procedure maintained the initial steady flow. It was expected that the linear interpolations in PATCH, TRUNET and ADDROW would introduce growing systematic errors. As might be expected, the flow field tended to be "smoothed" as the calculation proceeded in time. When linear interpolation is used to repeatedly approximate a function, the points which describe the function

tend to move toward the center of curvature of the function as shown in the following simplified sketch.



o initial points
x after first linear interpolation
● after second linear interpolation

The rate at which they move is proportional to the curvature. Of course, the numerical solution of the partial differential equations considered here is considerably more complicated, but the linear interpolation might be expected to have the same qualitative effect. This can be seen in Figure 22 where the body surface pressure is plotted. Note that the origin has been suppressed on the ordinate of the plot.

The time coordinates of the time surfaces for the four blunt body cases are given in Table II. When only one time is given TRUNET was used, and when a range of time is given the regular executive program was used.

The body surface pressure was found to be the quantity which changed the most rapidly with time. The velocity components as shown in Figures 23 and 24 did not change as rapidly in time. In fact they drifted only slightly from their initial values. The pressure on the middle ring of field points shown in Figure 25 has the same smoothing tendency. Note that the curvature in

Figure 25 is less than the curvature in Figure 22 on the body surface and that the rate at which the pressure falls with time is less than that in Figure 22. Also note that in all flow properties except the u component of velocity, which changes only slightly, the tendency is for the flow property to move toward its center curvature as would be expected if the linear interpolation were causing the drift.

In Figure 26 the pressure along the axis of symmetry from the body to the shock is plotted versus the x-coordinate. Again note that the origin on the ordinate has been extremely suppressed on this plot. It appears that the falling pressure on the body surface caused by the linear interpolation and the large "curvature" in pressure on the body surface actually introduces a mild expansion wave into the field. The drift due to the interpolation appears to reach such a magnitude that it is "picked up" and propagated through the field as an expansion wave.

The data from the two strip method of integral relations used here as initial data should be accurate to at least three digits. The two strip results are quite good as compared to the one strip results frequently used, which can be as much as 10% or more too low for the pressure on the body surface in the expansion region beyond the sonic line[51]. Thus, it cannot be reasoned that the unsteady method of characteristics is trying to adjust the initial steady flow data to a more accurate solution.

Perhaps a smaller net spacing would have reduced the drift

due to the linear interpolation (a short test case indicated that it did), but because the finer net results in a shorter time step (see equation (5.3)), the accuracy in calculating a given length of time for the flow would probably be about the same. Hence, it appears that higher order interpolation will be required to improve the accuracy of the calculation. The particular net size chosen was also a compromise to keep the total computer time required to carry out 50 time steps for each of the four blunt body cases at a reasonable value. Each case required about one and one half hours of IBM 7094 computation time to calculate the 50 steps.

It can be seen in Figures 22 and 25 that a small amount of asymmetry has developed in the field at the latest time steps. This was caused by a minor asymmetry in the organization and sequencing of the calculation by the executive program and TRUNET and can probably be eliminated by slight modifications to these programs. It was planned to call TRUNET at alternating time steps and to call ADDROW at every fourth step, but as can be seen from Table II this sequencing was not followed exactly. The sequence was disrupted because of human operational errors in the running of the programs. Because of the large amount of computer time involved, the cases were not rerun to follow the sequencing exactly and some asymmetry in the final data resulted. It will also be noted that the range of y over which the calculation was carried out decreased slightly as the calculation was continued in time. This was because the initial data did not continue out into the supersonic region far enough so that

the Mach cones leaned far enough downstream to make their intersections lie even further downstream. Thus, the total area covered by the calculation on each time surface did not increase, but decreased at such a slow rate that it did not introduce much of a restriction in the four blunt body cases considered.

A typical fluid particle entering the shock wave at about 30° above the axis of symmetry (near y = 0.8 in Figure 21) has an average velocity through the field of 2000 fps. The total time calculated in the 50 time steps was about 0.00025 seconds. The particle moved 0.5 feet in the 50 time step and passed about half way through the flow field being calculated. In order to watch five such particles pass through the field 500 time steps would have to be calculated. This agrees with the order of magnitude estimate made above for this particular net.

The second and third blunt body cases considered were specified by the following equations for the body surface:

for $0 \leq t' \leq UB$

$$B(x,y,t') = \left\{x^2\cos^2\alpha - xy\sin 2\alpha + y^2\sin^2\alpha\right\} \Big/ \left\{\frac{A-1}{2}\cos\left[\pi\left(\frac{t'}{UB}+1\right)\right] + \frac{A+1}{2}\right\}^2 + x^2\sin^2\alpha + xy\sin 2\alpha + y^2\cos^2\alpha - 1 = 0 \qquad (5.6a)$$

and for $t' > UB$

$$B(x,y,t') = x^2\left(\frac{\cos^2\alpha}{A^2} + \sin^2\alpha\right) + xy\left(1 - \frac{1}{A^2}\right)\sin 2\alpha + y^2\left(\frac{\sin^2\alpha}{A^2} + \cos^2\alpha\right) - 1 = 0 \qquad (5.6b)$$

where

$$A = 1.05$$
$$UB = 0.16 \text{ ft.}$$

and $\alpha = 0^\circ$ for the symmetric case and $\alpha = 20^\circ$ for the asymmetric case which results in an ellipse whose semi-major axis is 5% longer than its semi-minor axis. The semi-major axis lies along

the x-axis for the symmetric case and has a positive angle of attack of 20° for the asymmetric case. Note that the transition from the circle to the ellipse takes place smoothly using one half period of a cosine function.

Figure 27 shows the body surface pressure for the symmetric warping case. The pressure rises to a maximum on the eighth time surface when the body surface has its maximum velocity. It then falls to a level slightly below the initial pressure due to the slowing of the body motion. On the sixteenth surface the body motion has stopped. The pressure then tends to drift downward as in the steady flow in Figure 22. It appears that the steady flow downward drift has been superimposed on the basic unsteady flow and that, if the drift were removed with higher order interpolation, the stagnation point pressure would return to the same value it had initially as is to be expected.

The compression wave traveling from the body surface toward the shock wave can be seen in Figure 28. The peak pressure in the wave falls as the wave moves away from the body because of the two-dimensional nature of the wave. The wave speed relative to the fixed coordinate system as determined from the plot is approximately 2500 fps. initially and about 1500 fps. at the later time surfaces. Due to the low fluid velocity in the stagnation region, these values are approximately equal to the speed of sound in the field at these locations. For example, the speed of sound near the body surface is approximately 2200 fps. so that a compression wave only slightly stronger than a

sound wave has been obtained. Note that a small expansion region has developed in front of the compression wave. Of course this could not have been introduced by the body motion, but appears to arise from the pressure decrease caused by linear interpolation which was shown in Figure 26 in the steady flow and discussed above.

It can be seen from Figure 28 that the net is very coarse in the x-direction. It is almost too coarse to indicate properly the structure of the wave. It was necessary, however, to use this coarse a net and a fast body motion in order to carry out the calculation in a reasonable length of computer time.

In the 48 time steps calculated the wave has just reached the shock wave. The shock wave reacts properly to the wave in that it moves slightly toward the body when the small expansion region first hits and then moves more rapidly away from the body when the stronger compression wave strikes. From the decrease in amplitude of the wave in going from the body to the shock once, it appears that the transient time required for the flow to attain a steady state might be four times what has been calculated here. This agrees with the crude estimate of 200 time steps given at the first of this section.

The results of the asymmetric warping case are shown in Figures 29 and 30. The body surface pressure rises and falls in exactly the same way as in the symmetric case, but the distribution of the pressure along the body is now asymmetric. On the eighth time surface where the peak pressure is attained the

stagnation point has shifted up to a point 9° above the original axis of symmetry. After the body warping has stopped the stagnation point holds fairly steady at 5°.

The velocity vectors on a portion of the 16th and 34th time surfaces are shown in Figure 30. An idea of how far the compression wave has moved in the field by the 16th and 34th time surfaces can be obtained from Figure 28. In the 16th time surface of Figure 30 (a) the compression wave propagating toward the shock from the body can be clearly seen. Note that a region in which the gas is actually flowing upstream has developed next to the body. The asymmetry is evident in the compression wave between the wave front and the body. This asymmetry can also be seen in Figure 30 (b) where the compression wave has traveled three quarters of the distance from the body to the shock. The compression wave has spread and weakened so much that there is no longer an upstream flow region. The approximate location of the streamline which passes through the stagnation point has been sketched on the figure.

For the fourth blunt body case, the vertical oscillation is described by a body surface equation of the form

$$B(x, y, t') = x^2 + \left[y - A\{\cos Bt' - 1\}\right]^2 - 1 = 0 \qquad (5.7)$$

where

$$A = 0.02 \text{ ft.}$$

$$B = 25 \text{ ft.}^{-1}$$

The frequency of the motion is 7960 cycles per second and the reduced frequency is

$$k = \omega R / V_\infty = 10.28 \qquad (5.8)$$

This high frequency was chosen so that several periods of motion could be calculated in the 54 time steps considered.

The results of the calculation in the form of body surface pressure plots are given in Figure 31. The quasi-steady prediction of the maximum amplitude pressure distribution is shown in Figure 31(a). This quasi-steady distribution merely shifts instantaneously adjusting to the new effective free stream velocity obtained by vectorially subtracting the body velocity vector from the steady free stream velocity vector. It can be seen in the figure that the stagnation point pressure on the fourth and sixth time surfaces, which fall on either side of the maximum amplitude time, agree rather well with the quasi-steady value. The location of the stagnation point is at about a 50% greater y-coordinate than predicted by the quasi-steady result. There is also a marked asymmetry in the pressure distributions which is not predicted by the quasi-steady results.

No shift in the phase angle between the pressure oscillation and the motion of the body could be detected in the two cycles of motion calculated. The stagnation point pressure at the maximum pressure amplitude positions fell as the calculation was continued in time. This is probably the same effect noted in Figure 22 for the steady flow because the stagnation point pressure is falling at exactly the same rate of about 1½% per time step. The introduction of higher order interpolation should eliminate this downward drift.

Wedge Flow Check Cases

The first complete flow field to be analyzed was supersonic flow over a wedge of a perfect gas with $\gamma = 1.4$. The

executive program was a simplified form of the executive used for the blunt body and described in Chapter 4. Net configurations similar to the one shown in Figure 32 were used in approximately 250 separate flow field calculations.

The main purpose of the calculations was to check the various parts of the programs against known solutions. No attempt was made to carry out useful flow calculations. Note in Figure 32 that no attempt was made to extend the calculation to the point of contact of the shock wave with the wedge tip. This would require a separate program to calculate the special point which lies both on the shock wave and the wedge surface. This attached shock point could be easily calculated on a quasi-steady basis anyway, if it were needed. Because the flow is entirely supersonic, the area which can be calculated from the initial data decreases with each time step. One column of points running from the shock to the wedge is lost with each time step. Hence, only as many time surfaces as there are columns of points can be calculated. This limits the extent of the calculation, but this simple test case was only meant to check the complete procedure.

This flow field served as a very good check case. Various programming errors became apparent and many program improvements were determined. One major discovery was the numerical instability discussed above in Section 5.3. This instability was studied and eliminated using this wedge flow.

A few Mach three flows were calculated, but most of the

flows had a free stream Mach number of nine and a 30° shock wave angle. Hundreds of runs were made either with the wedge surface held steady in time or impulsively moving it up or down at right angles to the free stream velocity. A few runs were made with changing free stream conditions or with an irregularly shaped shock wave. Most of these calculations were made while the programs were being debugged and improved so that only a few of the last calculations performed can be considered meaningful results.

It was felt that rather than repeat many of the wedge flow calculations after the programs worked well, more could be learned by proceeding to more interesting and complicated flow fields. Hence, only the results of the last few wedge flow calculations are presented here. When the wedge surface was held steady, the steady flow results held constant to six and seven digits. This was to be expected because the flow properties were constant in the field between the shock and the wedge surface. No outer loop iteration was necessary and only the coordinates of new points had to be determined. The results of four calculations in which the wedge surface was impulsively set into motion upward at $t' = 0$ are presented in Figure 33. The wedge was moved upward at four different velocities. This gave rise to a compression wave which propagated from the wedge surface out toward the shock wave. This impulsive upward velocity is equivalent to an instantaneous increase in the wedge angle. The instantaneous acceleration also corresponds to a corner or discontinuity

on the body surface. The flow about bodies with discontinuous surfaces or corners cannot be calculated with the present programs as was pointed out in Chapter 3. Small corners should introduce only small errors in the flow calculation, but these errors will grow with the size of the discontinuity. The results of these wedge flows can be used to determine the size of the body surface discontinuity which can be tolerated without introducing large errors into the flow calculation.

In Figure 33(a) the pressure on the wedge surface for steady flow calculated with the oblique shock equations (Reference No. 50) is plotted versus the pressure on the wedge surface at the latest time that could be calculated before no more points were available to continue the calculation. The disagreement of the pressures increases with the instantaneous wedge angle change for two reasons. One is that, for larger impulsive angle changes which correspond to body surface discontinuities, the finite difference approximations become less accurate as mentioned above. Actually, very large angle changes could lead to the formation of shock waves. A tendency for the compression waves to steepen was noted in some of the calculations even though the formation of a shock cannot be accurately calculated with the present programs. The second reason for the increasing disagreement is that the flow field had not settled to a completely steady state at the time of the last step in the calculation. It appears that the body surface discontinuity is too large for the fourth run which has an instantaneous angle change of 2.37°.

Hence, the area to the right of the shaded region probably corresponds to a body surface discontinuity which is too large.

Figure 33(b) compares the Mach number behind the shock from the oblique shock equations with the Mach number from the numerical calculations. The Mach number was obtained indirectly from the numerical results by using p and s to determine a which was then divided into the velocity magnitude. These Mach number results seem to be a little less accurate than the pressure results, but seem to have the same trend.

Figure 34 shows a typical time history of the pressure along the wedge surface. These results are from case number 3 of Figure 34. The variation of pressure along the surface on the 8th, 12th, and 18th time surface is well within the accuracy to be expected for the convergence test values of 0.001 used in the calculation. The variation on the 4th surface is slightly larger than might be expected, but it is possible that the abrupt nature of the instantaneous vertical motion caused some initial spurious transient. It is also possible, however, that the variation results from a combination of the organization used in the executive program and the iterations in the various subroutines so that the results are not as accurate as the single convergence test value of 0.001 might indicate.

5.5 Conclusions

The solution of unsteady two-dimensional flow fields by the method of characteristics is feasible and practical at

the present time utilizing presently available high speed electronic digital computers. The solution of the more difficult problems involving nonequilibrium thermodynamics or three-dimensional unsteady flows is marginal with existing computers but should become completely feasible with the next generation of computers which will appear shortly.

The problems being considered here involve large amounts of input and output data. For example, the output data from the four blunt body cases discussed in the previous section was in the form of approximately one half million, eight digit numbers. The handling and analysis of this data was by no means a simple task. The development of methods and devices for quickly handling large amounts of data and displaying it quickly in graphical or other useful forms is mandatory.

The introduction of higher order interpolation into the programs is absolutely necessary in order to obtain high enough accuracy to solve engineering and scientifically useful problems. The preliminary conclusion of Chapter 1 that the method characteristics is potentially the most accurate of the three methods considered there has had no doubt cast upon it by the results of this study. Quite to the contrary, it appears that rather encouraging results were obtained despite the very crude linear interpolations used to obtain the results.

The utility of the method of characteristics applied to multi-dimensional flow fields is obvious. In the course of trying to decide what flow fields should be calculated as typical example

cases, so many came to mind that the choice was difficult. Many interesting unsteady hypersonic flows, which were the motivation of this study, can be calculated with the method. The vertical oscillation of the blunt body presented in the previous section is just one interesting example. Other types of oscillations or transient motions can easily be handled with the method. Interesting steady flows might be determined by starting from some known steady flow and passing through a specified transient to a second more interesting and perhaps previously unknown steady flow. The symmetric and asymmetric warping results presented in the previous section give encouraging indications that this technique is feasible and practical. Many other interesting applications of the method can be easily thought of by merely letting the mind wander.

One limiting aspect of the method is the large amount of initial data required to start the solution. Many theories are available to give certain specific details of flow fields, but the complete details of entire flow fields are seldom determined and hardly ever published. For this reason the application of the method might be somewhat limited, but there is the intriguing possibility of specializing the general method in certain cases so that it could determine its own initial data. For example, the initial conditions for a three-dimensional unsteady flow might be obtained by using a specialized form of the general method to solve a three-dimensional steady supersonic flow.

CHAPTER 6

EXTENSIONS AND FUTURE DEVELOPMENTS

6.1 Immediate Improvements to Existing Programs

The existing programs listed in Appendix C are far from being optimized with respect to operating time and storage requirements. A few improvements which can be made to these programs, are listed in this section. This is not an exhaustive list of improvements, but includes those changes which appear advantageous from results obtained through operation of the programs, up to the present time.

The iteration loops in the subroutines could be improved by using both residual tests and convergence test criteria in the form of (4.1). Then, if the convergence test value were too small, the residual tests could still release the calculation from the loop. Conversely, if the convergence test values did not require that the residuals be reduced to zero to eight decimal digits, some computation time could be saved by releasing the calculation from the loop as soon as the convergence test criteria were satisfied. Of course, residual tests can be applied to only those loops which are solving algebraic equations, because residuals do not exist for the loops which are integrating differential equations. As pointed out in Section 3.2, the inner iteration loops in FLDPT, BDYPT, and SHKPT can be eliminated by more closely utilizing the third order Richmond iteration scheme. This might reduce the computation time required to

determine the coordinates.

The executive program and TRUNET can be combined into one program with a length slightly longer than TRUNET's. In the process of writing this combined program, the possibility of using higher order interpolation should be investigated. More labeling of the printed output could also be included in the program, together with more options for suppressing or obtaining the large quantities of output.

The PATCH subroutine can be extensively improved. First, the unsuccessful option for locating body and shock points discussed in Section 4.3 can be removed, and the size of the program substantially reduced. The iteration loop tests can be improved as discussed in the second paragraph of this section. Higher order interpolation could be substituted for the linear interpolation currently used in the subroutine.

The convergence criterion for the components of the shock wave normal vector could be studied and perhaps improved. In the current SHKPT subroutine, the convergence test value has been arbitrarily set at ten times the outer loop convergence test value.

Finally, the executive and also, the TRUNET programs could be much more efficiently written in the FAP computer language because they are basically manipulative, rather than arithmetic programs. Of course, probably all of the programs could be improved somewhat if written by skilled programmers in FAP.

6.2 Future Developments

As mentioned in Chapters 1 and 4, whenever limitations were introduced into the problem being considered, they were done in such a way that they could be most easily removed at a later time. For instance, the nonorthogonal Thornhill coordinate transformation discussed in Section 3.2 could have been used to simplify the two-dimensional unsteady flow calculated here. But, it was not used because no generalized transformation of this type is available to simplify the three-dimensional unsteady problem. In this section, developments are discussed which can be added to the programs in the future. It is felt that no fundamental changes will be necessary in the programs, or the analysis to carry out these extensions.

The programs can be extended to consider three-dimensional unsteady flows when the newest computers, which are being built at the present time, are available. These machines will have magnetic core storage units of 130,000 to 260,000 words, and will make the solution of multi-dimensional problems practical. The addition of a single space dimension to the programs presented here should be rather simply accomplished. The double iteration process to solve the four Mach hypercone equations, (3.4a), was programmed and debugged to check the feasibility of the process for four independent variables. The program was written in such a way that the equations for three Mach cones in two-dimensional unsteady flow could also be solved. The program worked very well, but it was concluded that its

length was increased to a large extent because both three and four independent variable solutions could be determined, rather than just one or the other.

Changing the programs to consider equilibrium, real gases is a rather simple process, as mentioned in Chapters 3 and 4. The only practical drawback might be prohibitive increases in computation time. This too, could be overcome by the new computers to be available shortly which will use multiple processing methods and equipment to decrease the time required to complete a given calculation.

Nonequilibrium flows can also be attempted with these higher speed machines. The inclusion of the integration of the species continuity equation along the streamline, will require major modifications to the existing programs, but experience being gained at the present time in calculating two-dimensional steady nonequilibrium flows, can be utilized to facilitate the modification. The increased storage capacity and higher speed of future machines should make the calculation of this most general flow possible.

In considering more general problems, points falling on interior shock waves, contact surfaces and expansion waves must also be calculated. These might result from bodies which have corners and other discontinuities. Procedures can be developed to calculate such points (see Butler[28], for example). Subroutines similar to the FLDPT, BDYPT, and SHKPT will have to be written for these more complicated situations. These subroutines appear

at least feasible, but will require an extensive programming effort to write a practical set of subroutines.

In the more distant future, it is possible that a universal inviscid flow program could be developed which solved the three-dimensional, unsteady, inviscid, nonequilibrium flow of an arbitrary mixture of gases about a general body. The program could be written in such a way that it could also do problems which are special simplified cases of the most general flow. At the present time, there are many roadblocks to writing such a program in a practical form, but there appear to be no fundamental reasons why such a program could not be created.

It is possible that some day, this program could be combined with other programs that compute boundary layers or other portions of the flow where viscosity, heat conduction, and diffusion might be important. Electromagnetic effects might also be included. At present, some people are thinking about using computers as design tools, so that designers or investigators working at new and yet to be developed input-output devices, could interact more directly with computers in the design process. Then, for example, an aircraft or spacecraft designer could use the general flow field programs discussed here, together with structural stress analysis programs, heat transfer programs, control systems programs, etc. to work at a computer console and quickly and efficiently design the vehicle. The programs would be written such that the output of one would be used as input for several others and vice versa. Thus, they could be used in an iterative manner

to optimize the design. After the vehicle is designed, it is conceivable that it could be built and assembled by automated machine tools which are directly connected to the computer used in the design process.

It appears that there are vast possibilities for the utilization of high speed electronic digital computers and numerical solution methods in solving complicated nonlinear problems. This is especially true when very general boundary conditions and complicated physical processes must be considered.

APPENDIX A

A Brief Review of the Theory of Characteristics

The mathematical theory of characteristics for hyperbolic partial differential equations is well developed and is considered classical. The complete theory will not be reiterated here. Only those properties and equations utilized in developing the particular numerical method of concern to this analysis are presented. For the complete mathematical theory, the reader is referred to Chapter 6 of the second volume of Courant and Hilbert[37]. The theory applied to the flow of a compressible fluid is discussed by von Mises[38] (pp. 100-134).

Consider a set of quasi-linear first-order partial differential equations which can be written in the form*

$$A_{ijk} \frac{\partial u_k}{\partial x_i} + B_j = 0 \qquad \text{(A.1)}$$

Quasi-linear means that the highest-order derivatives appear to only the first degree. Note that considering first-order equations is not a restriction because higher-order sets of equations can be reduced to first-order by defining new variables. In (A.1), u_k is a general dependent variable, and x_i an independent variable. A_{ijk} and B_j are functions of u_k

- - - - - - - - - - - - - - - - - -

* The index summation notation of Cartesian tensors is employed.

and x_i only. Next, (A.1) is transformed to a new set of coordinates, β_m.

$$A_{ijk} \frac{\partial u_k}{\partial \beta_m} \frac{\partial \beta_m}{\partial x_i} + B_j = 0 \qquad \text{(A.2)}$$

Consider a manifold of one less dimension than the parent space, e.g., for three independent variables, a two-dimensional surface. Let the manifold be given by

$$\beta_1 = \text{constant} \qquad \text{(A.3)}$$

upon which u_k and all $\partial u_k / \partial \beta_m$ except $\partial u_k / \partial \beta_1$ are given. The $\partial u_k / \partial \beta_1$ are to be determined by using equations (A.2). If (A.2) are considered a complete set of algebraic equations in the $\partial u_k / \partial \beta_1$, they can be solved for $\partial u_k / \partial \beta_1$ only if

$$\det \left\{ A_{ijk} \frac{\partial \beta_1}{\partial x_i} \right\} \neq 0 \qquad \text{(A.4)}$$

The manifolds for which (A.4) is not true are the characteristic manifolds, e.g., for three independent variables — a characteristic surface. Thus

$$\det \left\{ A_{ijk} \frac{\partial \beta_1}{\partial x_i} \right\} = 0 \qquad \text{(A.5)}$$

is termed the characteristic equation and defines the characteristic manifolds. Note that when u_k and $\partial u_k / \partial \beta_m$ $(m \neq 1)$ are given on the characteristic manifold, $\partial u_k / \partial \beta_1$ cannot be

determined from (A.2) because of (A.5). This is one property of the characteristic manifolds. There are other properties which could be and have been used to define them, but these all lead to the same characteristic equation, (A.5).

Next, consider a linear combination of the equations (A.2), which can be written in the form

$$\lambda_j A_{ijk} \frac{\partial u_k}{\partial \beta_m} \frac{\partial \beta_m}{\partial x_i} + \lambda_j B_j = 0 \qquad \text{(A.6)}$$

or this can also be written in the form

$$C_k \frac{\partial u_k}{\partial \beta_1} + D_k \frac{\partial u_k}{\partial \beta_2} + E_k \frac{\partial u_k}{\partial \beta_3} + F_k \frac{\partial u_k}{\partial \beta_4} + \ldots + G = 0 \qquad \text{(A.7)}$$

where the definition of the new symbols is

$$C_k = \lambda_j A_{ijk} \frac{\partial \beta_1}{\partial x_i} \qquad \text{(A.8a)}$$

$$D_k = \lambda_j A_{ijk} \frac{\partial \beta_2}{\partial x_i} \qquad \text{(A.8b)}$$

$$E_k = \lambda_j A_{ijk} \frac{\partial \beta_3}{\partial x_i} \qquad \text{(A.8c)}$$

$$F_k = \lambda_j A_{ijk} \frac{\partial \beta_4}{\partial x_i} \qquad \text{(A.8d)}$$

$$\vdots \qquad\qquad \vdots$$

$$G = \lambda_j B_j \qquad \text{(A.8e)}$$

On the characteristic manifold given by

$$\beta_1 = \text{constant}$$

(A.5) holds, so λ_j can be found such that

$$C_k = \lambda_j A_{ijk} \frac{\partial \beta_1}{\partial x_i} = 0 \qquad \text{(A.9)}$$

Then, on the characteristic manifold, equation (A.7) has the form

$$D_k \frac{\partial u_k}{\partial \beta_2} + E_k \frac{\partial u_k}{\partial \beta_3} + F_k \frac{\partial u_k}{\partial \beta_4} + \ldots + G = 0 \qquad \text{(A.10)}$$

where D_k, E_k, F_k and G are given by (A.8b) through (A.8e) and λ_j is determined from (A.9). Equation (A.10) is called the compatibility equation and has the useful property that derivatives normal to the characteristic manifold do not appear. In general, there might be more than one compatibility equation for a particular characteristic manifold, but there must be at least one such equation. The compatibility equations are basic for the method of characteristics.

APPENDIX B

A Note on Stability

Consider a set of first order hyperbolic partial differential equations of the form

$$\frac{\partial \vec{f}}{\partial t} = \sum_{k=1}^{m} \bar{\bar{A}}^k \frac{\partial \vec{f}}{\partial \alpha^k} \tag{B.1}$$

where $\vec{f}$ is a vector made up of n dependent variables, α^k are m space coordinates and $\bar{\bar{A}}^k$ are real constant, coefficient matrices Courant, Friedrichs and Lewy[52] determined that a necessary condition for convergence of a difference scheme for this set of equations is that the domain of dependence of the diffrence scheme must contain the domain of dependence of the differential equation. Hahn[53], using the work of Lax[54], showed that for simplicial difference schemes (i.e., schemes that use a minimum number of points in the initial surface to determine a new point) the Courant-Friedrichs-Lewy condition is sufficient as well as necessary for convergence, and therefore, also for stability.

The linearized equations for inviscid nonequilibrium three-dimensional unsteady flow can be put in the form of equation (B.1). The equation for one- and two-dimensional unsteady flow and two- and three-dimensional supersonic steady flow also have this form. Hence, the stability and convergence criterion mentioned in the above paragraph applies, at least locally (because of the linearization), to these types of flows.

It is possible to remove the restriction that the criterion applies only locally because of the linearization if the equations and the true solution possess a certain degree of smoothness. Strang[55] has shown that, if the equations and their solution possess enough continuous derivatives, then the convergence of the full nonlinear equations depends on the stability of the linearized equations. The existence of the continuous derivatives of the solution cannot be proved before the solution is known, but it might be expected that for "reasonably well behaved" physical problems, the solution will possess the required degree of smoothness.

This stability criterion can be applied to the basic finite difference networks proposed for problems with three independent variables, and discussed in Section 3.1. The tetrahedral characteristic line network of Figure 5 is found to be unstable when the stability criterion is applied. A good approximation to the domain of dependence of point P_4 in the initial surface, is a circle through points P_1, P_2, and P_3. This is the domain of dependence for the partial differential equations. The domain of dependence of the difference net is the triangle with vertices at P_1, P_2, and P_3 (formally termed the convex hull of the difference scheme). The domain of dependence of the difference scheme does not contain the domain of dependence of the differential equations, and therefore, does not fulfill the stability criterion.

Physically, the stability criterion seems very logical, because it is not hard to imagine that in certain areas of a flow field, points lying within the circle but outside the triangle circumscribed by the circle, could have a large influence on the new point being calculated. However, in this network, effects of points outside the triangle are not taken into account in the numerical calculation, so that the solution could be erroneous.

To insure stability, a modification to the tetrahedral characteristic line network (shown in Figure 6) is proposed. Starting with the original base points P_1, P_2, and P_3, intermediate base points P_{12}, P_{23}, and P_{13} are determined by inscribing a circle within the triangle with vertices at the base points. The points of tangency between the circle and the sides of the triangle are chosen as intermediate base points and the flow properties at these points are determined by interpolation. The intermediate base points are used in exactly the same way as the base points, P_1, P_2, and P_3, were used in the original network. This modified network fulfills the stability criterion, in that the circular domain of dependence of the differential equations falls within the triangular domain of dependence of the difference scheme. If the same base points are used to calculate a new point with both the modified and unmodified networks, the step size from the initial surface to the new point is smaller for the modified network. Thus, the use of the modified network requires a larger number of calculations to do a given problem, but it is well known that, in standard finite

difference techniques, decreasing the step size has a stabilizing influence.

A numerical calculation has been made of the flow of a perfect gas about a circular cylinder with its axis normal to a Mach 5 free stream. Both the modified and unmodified networks were used to calculate the flow with the body surface held constant in time. It was found that the pressure along the axis of symmetry was the most sensitive indicator of the onset of instability. Figure 35 shows plots of the pressure ratio versus the coordinate parallel to the free stream velocity for points closest to the axis after various time steps in the calculation. Network A is the unmodified tetrahedral characteristic line network, while Network B is the modified form of the same net. Note that these points are not located exactly on the axis of symmetry and do not have exactly the same space coordinates at each step, so that the magnitudes of the pressure ratio should not be exactly the same on each plot, but they do very graphically indicate the onset of instability. The instability with the unmodified network had grown so large by the eighth step that calculation could not be continued, whereas no instability has been detected while using the modified network.

The tetrahedral characteristic surface network is also simplicial and is found to be stable when the stability criterion is applied. Tsung[29] utilized this network and indeed, found no instability in his calculations.

The last three networks discussed in Chapter 3 and shown in Figures 8 to 10, are not simplicial and the stability criterion cannot be formally applied. The physical argument given in the fifth paragraph of this Appendix would still seem to apply to these networks, however. Hence, application of the criterion would seem to give some insight into the stability of the net, even though it does not formally apply.

It appears that these three networks might be unstable, if only the local net configuration is considered. However, Moretti, et.al.[30] has utilized the network of intersections of reference planes with characteristic surfaces, and Butler[28], the pentahedral bicharacteristic line network, and they encountered no instability in their calculations. Closer examination of the networks reveals that in each of the calculations, the base points in the initial surface must be moved about in the initial surface as the solution at the new point is obtained by iteration. This requires that the properties at the base points must be obtained by interpolation in the initial surface. The interpolation schemes used by Moretti and Butler are such that they effectively increase the domain of dependence of the difference scheme to the extent that it contains the domain of dependence of the differential equations. Hence, the overall schemes could be stable.

Now, in applying the stability criterion to the problem of four independent variables, four base points in an initial hypersurface must be considered. A net utilizing four base points is simplicial so the criterion can be applied. The four

base points define a tetrahedron in the initial hypersurface which is the domain of dependence of the finite difference scheme. The domain of dependence of the partial differential equations is the intersection of the characteristic hypersurface (the Mach hyperconoid) with the initial hypersurface. If the initial hypersurface is a t' = constant hypersurface, the intersection is a sphere or can be closely approximated with a sphere for most reasonable initial hypersurfaces. Hence, if the domain of dependence of the finite difference scheme is to contain the domain of dependence of the differential equations, the sphere must be contained within the tetrahedron. The modified tetrahedral characteristic line network, shown in Figure 11 and discussed in detail in Chapter 3, fulfills the above requirement and therefore, should give stable results.

Previously, it has been tacitly assumed in characteristic calculations involving three independent variables that integration on the characteristic surface, or at least along bicharacteristic lines, assures the convergence and stability of the process. This is not the case. Care must be taken in choosing a net configuration which is stable and convergent. The Courant-Friedrichs-Lewy condition can be employed as a test for stability and convergence of a finite difference network which is simplicial, and it can be argued, physically, that this provides a necessary condition for the stability of networks which are not simplicial.

APPENDIX C

Program Flow Diagrams and Listings

It is felt that exact program listings are much more useful than detailed flow diagrams in attempting to duplicate or extend existing computer programs. Detailed flow diagrams still require much time and effort to translate them into working computer programs, and because of their detail, they are not easy to use in attempting to understand the broad pattern in the program. Listings, on the other hand, require only keypunching to obtain working programs. For this reason, only a few very general flow diagrams for the FLDPT, BDYPT, and SHKPT subroutines are included here to give a general idea of the program organizations. Exact copies of the listings are presented to make the programs easily reproducible. These listings are not in the neatest possible form, but have been left in the exact form to which they evolved in the debugging process. The listings have been left in this form to eliminate the possibility of introducing transcriptions errors in a process of putting them in a neater format.

The M.I.T. Computation Center has altered its system, so that READ, PRINT, and PUNCH FORTRAN statements read and write tapes and thus all input and output is done off-line. XSIMEQF is a FAP coded subprogram available on the Computation Center Library tape which solves the matrix equation

$$[A]\,[X] = [B]$$

where A and B are given and X is determined. EXIT is the subroutine called to accomplish a normal termination of the program execution.

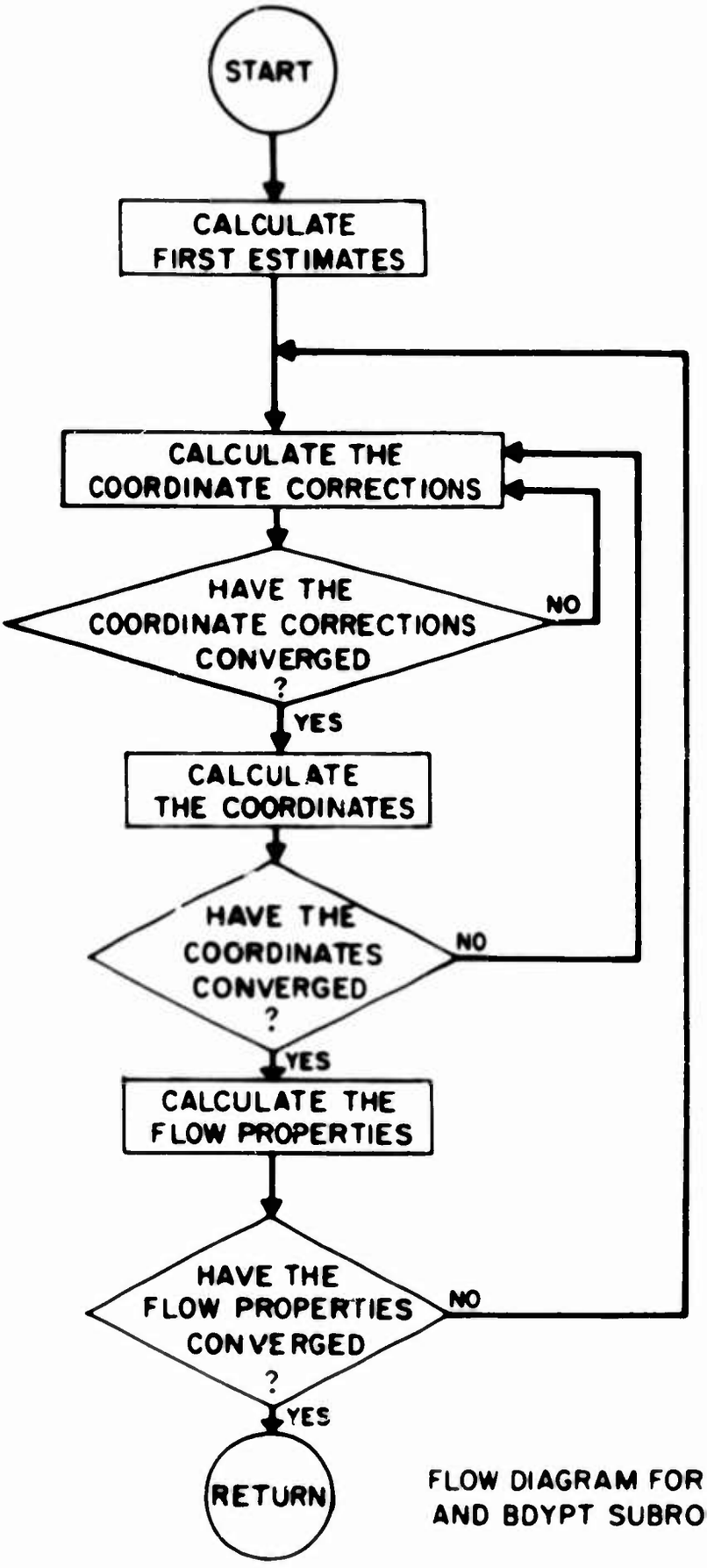
The flow diagrams and listings follow in the following order:

Diagrams

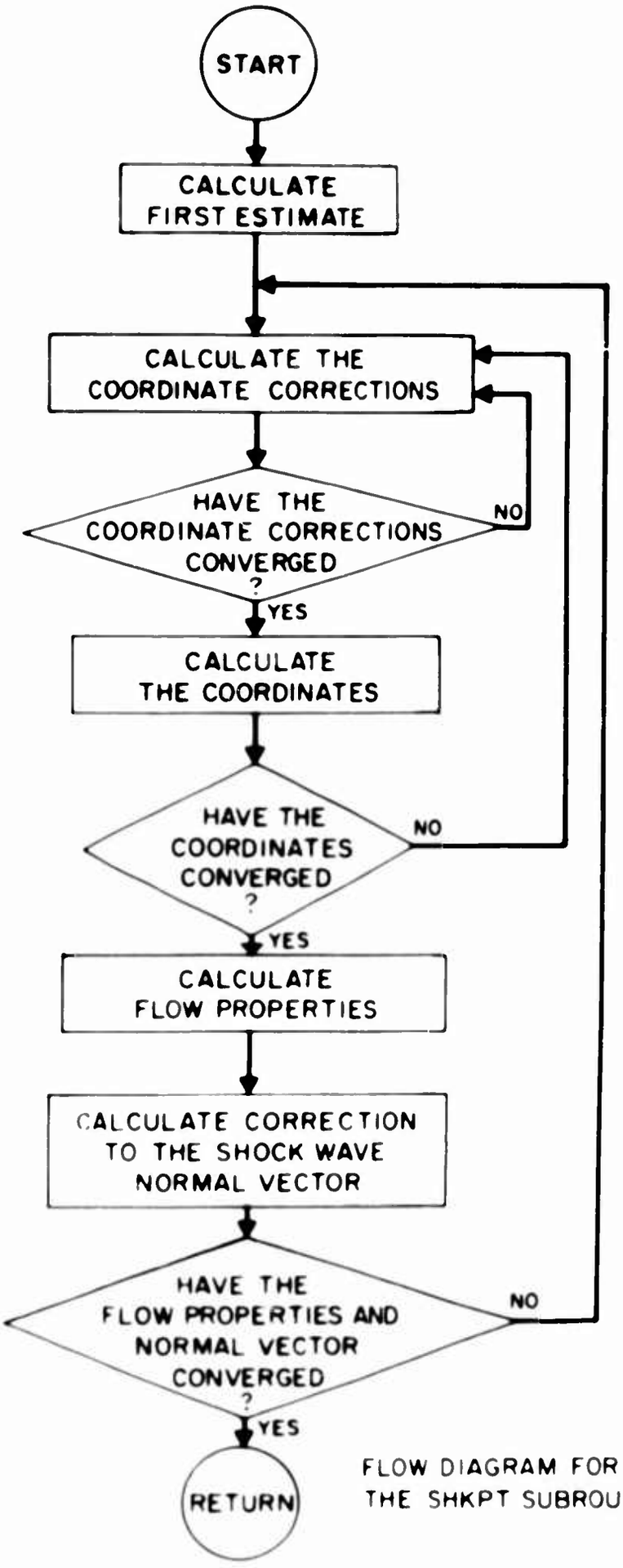
1. Flow Diagram for the FLDPT and BDYPT Subroutines
2. Flow Diagram for the SHKPT Subroutine

Listings

1. Executive Program
2. SCAN
3. ADDROW
4. TRUNET
5. TINTER
6. PATCH
7. FLDPT
8. SHKPT
9. BDYPT
10. Several XCHANG's
11. General form of XCHANG
12. XINTER
13. ADET
14. ROEDET
15. THETA



FLOW DIAGRAM FOR THE FLDPT AND BDYPT SUBROUTINES



FLOW DIAGRAM FOR THE SHKPT SUBROUTINE

```
C     EXECUTIVE TO CALCULATE 2-D UNSTEADY FLOW OVER BLUNT BODIES
C     USING SUBROUTINES TRUNET AND PATCH
C     THIS PROGRAM OBTAINS ITS DATA FROM TAPE (PHYSICAL 89, LOGICAL 10)
C     THIS EXECUTIVE REQUIRES ACCESS TO SUBROUTINES PATCH TRUNET
C     BDYPT SHKPT AND FLDPT
      DIMENSION FLPTS1 (90 , 20, 7), BDPTS1 (90 , 7), SKPTS1 (90 , 10),
     1 X(9), Y(9), T(9)
     2, U(9), V(9), P(9), S(9), BCOEF(4)
      COMMON X, Y, T, U, V, A4, UCAP, P, S, ROE4, DXTEST, XTEST, VTEST,
     1BCOEF, ENX4, ENY4, ENT4, ENR5, ENV5, ENT5, FLPTS1,BDPTS1,SKPTS1,
     2TEST,UPLUS,VPLUS,PPLUS,SPLUS
C     IND = 1 CAUSES NO INTERPOLATION ON FIRST STEP.  IND = 2 CAUSES
C     INTERPOLATION ON FIRST STEP.  ITNOTS INDICATES THE NUMBER OF
C     NORMAL STEPS BETWEEN ITERATIONS.  ITNOTS = 0 CAUSES INTERPOLATION
C     AT EACH STEP.  ITIMSR INDICATES THE TOTAL NUMBER OF STEPS.
C     NUM INDICATES THE LOCATION OF THE DATA ON LOGICAL TAPE 10.
C     PHYSICAL 89.            NPRINT=1 CAUSES PRINT OUT OF INITIAL
C     SURFACE, NPRINT=2 SUPRESSES PRINTING OF THE INITIAL SURFACE.
C     INDADD = 1 ADDS KADD ROWS TO M BY INTERPOLATION THE FIRST TIME
C     TRUNET IS CALLED.  INDADD = 2 CAUSES NO INTERPOLATION THE FIRST
C     TIME.  KADD ROWS ARE ADDED EVERY TIME ADDROW IS CALLED.
C     INADST INDICATES THE NUMBER OF TRUNET CALLS WITHOUT ADDROW BETWEEN
C     ADDROW CALLS.
      READ 1,IND,ITNOTS,ITIMSR,NUM,NPRINT, INDADD, INADST, KADD
      PRINT1,IND,ITNOTS,ITIMSR,NUM,NPRINT, INDADD, INADST, KADD
      REWIND 10
      READ 2, UPLUS, VPLUS, PPLUS, SPLUS
      PRINT2, UPLUS, VPLUS, PPLUS, SPLUS
      READ 2, UCAP, DXTEST, XTEST, VTEST
      PRINT2, UCAP, DXTEST, XTEST, VTEST
      READ  2, TEST
      PRINT 2, TEST
      READ TAPE 10, NUM1
      IF (NUM - NUM1) 801, 800, 800
  800 PRINT 802
  802 FORMAT (54H TRIED TO READ TAPE BEYOND POINT WHERE DATA IS WRITTEN)
      CALL EXIT
  801 CALL SCAN (NUM)
      READ TAPE 10,M,N
      BACKSPACE 10
      PRINT 1,M,N
      MINUS = M - 1
      NPLUS = N + 1
      INC = 0
      INC1 = 0
      ITIMS = 0
        READ TAPE 10,M,N,(((FLPTS1(I,J,K),K=1,7),J=2,NPLUS),I=1,M),
     1((BDPTS1(I,K),K=1,7),I=1,M),((SKPTS1(I,K),K=1,10),I=1,M)
      REWIND 10
    1 FORMAT (8I3)
    2 FORMAT(4F20.9)
    3 FORMAT (9E13.8)
    4 FORMAT (7E16.8)
      GO TO(200,201),NPRINT
  200  PRINT 202
  202 FORMAT(44H INITIAL DATA - FIELD, SHOCK AND BODY POINTS)
      PRINT4,(((FLPTS1(I,J,K),K=1, ),J=2,NPLUS),I=1,M)
      PRINT 100,((SKPTS1(I,K),K=1,10),I=1,M)
      PRINT 4,((BDPTS1(I,K),K=1, 7),I=1,M)
      PRINT 209
  209 FORMAT(20H END OF INITIAL DATA)
  201 GO TO (90, 93), IND
   90 IF (ITNOTS) 92, 93, 92
   92 DO 41 IL=1,ITNOTS
   31 IF (M - 2) 29, 29, 32
C     THERE ARE TWO TYPES OF TIME SURFACES COMPUTED BY THIS EXECUTIVE
C     AN ODD NUMBERED TIME SURFACE CONTAINS ONLY FIELD POINTS
C     AN EVEN NUMBERED TIME SURFACE CONTAINS FIELD,SHOCK, AND BODY PTS.
C     THE COMPUTATION FOR THE ODD NUMBERED SURFACE BEGINS HERE
   32 DO 16 I=1,MINUS
      DO 26 K=0,1
      IK = I + K
      X(K+1) = FLPTS1 (IK , 2, 1)
      Y(K+1) = FLPTS1 (IK , 2, 2)
      T(K+1) = FLPTS1 (IK , 2, 3)
      U(K+1) = FLPTS1 (IK , 2, 4)
      V(K+1) = FLPTS1 (IK , 2, 5)
      P(K+1) = FLPTS1 (IK , 2, 6)
   26 S(K+1) = FLPTS1 (IK , 2, 7)
      X( 3 ) = SKPTS1 (I ,1)
      Y( 3 ) = SKPTS1 (I ,2)
      T( 3 ) = SKPTS1 (I ,3)
      U( 3 ) = SKPTS1 (I ,4)
      V( 3 ) = SKPTS1 (I ,5)
      P( 3 ) = SKPTS1 (I ,6)
      S( 3 ) = SKPTS1 (I ,7)
      CALL PATCH (1, TEST)
      CALL FLDPT
      FLPTS1 (I, 1, 1) = X(4)
      FLPTS1 (I, 1, 2) = Y(4)
      FLPTS1 (I, 1, 3) = T(4)
      FLPTS1 (I, 1, 4) = U(4)
      FLPTS1 (I, 1, 5) = V(4)
      FLPTS1 (I, 1, 6) = P(4)
      FLPTS1 (I, 1, 7) = S(4)
      DO 15 J=2, N
      DO 16 K=0,1
      JK = J + K
      X(K+1) = FLPTS1 (I, JK , 1)
      Y(K+1) = FLPTS1 (I, JK , 2)
      T(K+1) = FLPTS1 (I, JK , 3)
      U(K+1) = FLPTS1 (I, JK , 4)
      V(K+1) = FLPTS1 (I, JK , 5)
      P(K+1) = FLPTS1 (I, JK , 6)
   16 S(K+1) = FLPTS1 (I, JK , 7)
      X(3) = FLPTS1 (I+1,J+1,1)
      Y(3) = FLPTS1 (I+1,J+1,2)
      T(3) = FLPTS1 (I+1,J+1,3)
      U(3) = FLPTS1 (I+1,J+1,4)
      V(3) = FLPTS1 (I+1,J+1,5)
      P(3) = FLPTS1 (I+1,J+1,6)
      S(3) = FLPTS1 (I+1,J+1,7)
      CALL PATCH (1, TEST)
      CALL FLDPT
      FLPTS1 (I,J  ,1) = X(4)
      FLPTS1 (I,J  ,2) = Y(4)
      FLPTS1 (I,J  ,3) = T(4)
      FLPTS1 (I,J  ,4) = U(4)
      FLPTS1 (I,J  ,5) = V(4)
      FLPTS1 (I,J  ,6) = P(4)
   15 FLPTS1 (I,J  ,7) = S(4)
      X(  1) = FLPTS1 (I  , NPLUS, 1)
```

```
      V(  1) = FLPTS1 (I  , NPLUS, 2)
      T(  1) = FLPTS1 (I  , NPLUS, 3)
      U(  1) = FLPTS1 (I  , NPLUS, 4)
      V(  1) = FLPTS1 (I  , NPLUS, 5)
      P(  1) = FLPTS1 (I  , NPLUS, 6)
      S(  1) = FLPTS1 (I  , NPLUS, 7)
      DO 17 K=0,1
      IK = I + K
      X(K + 2) = BDPTS1 (IK, 1)
      Y(K + 2) = BDPTS1 (IK, 2)
      T(K + 2) = BDPTS1 (IK, 3)
      U(K + 2) = BDPTS1 (IK, 4)
      V(K + 2) = BDPTS1 (IK, 5)
      P(K + 2) = BDPTS1 (IK, 6)
17    S(K + 2) = BDPTS1 (IK, 7)
      CALL PATCH (1, TEST)
      CALL FLDPT
      FLPTS1(I,NPLUS,1)= X(4)
      FLPTS1(I,NPLUS,2)= Y(4)
      FLPTS1(I,NPLUS,3)= T(4)
      FLPTS1(I,NPLUS,4)= U(4)
      FLPTS1(I,NPLUS,5)= V(4)
      FLPTS1(I,NPLUS,6)= P(4)
14    FLPTS1(I,NPLUS,7)= S(4)
      M = MINUS
      MINUS = M - 1
20    PRINT 4,(((FLPTS1(I, J, K), K=1,7), J=1,NPLUS), I=1,M)
C     THE CALCULATION OF AN EVEN NUMBERED SURFACE BEGINS HERE
C     FIRST ALL SHOCK POINTS ARE COMPUTED
      PRINT 5
5     FORMAT (16H STARTING SHKPTS)
      DO 21 I=1,MINUS
      DO 33 J=1,2
      K1 = I + J - 1
      X(J  ) = FLPTS1 (K1 , 1, 1)
      Y(J  ) = FLPTS1 (K1 , 1, 2)
      T(J  ) = FLPTS1 (K1 , 1, 3)
      U(J  ) = FLPTS1 (K1 , 1, 4)
      V(J  ) = FLPTS1 (K1 , 1, 5)
      P(J  ) = FLPTS1 (K1 , 1, 6)
      S(J  ) = FLPTS1 (K1 , 1, 7)
      X(J+2) = SKPTS1 (K1, 1)
      Y(J+2) = SKPTS1 (K1, 2)
      T(J+2) = SKPTS1 (K1, 3)
      U(J+2) = SKPTS1 (K1, 4)
      V(J+2) = SKPTS1 (K1, 5)
      P(J+2) = SKPTS1 (K1, 6)
33    S(J+2) = SKPTS1 (K1, 7)
      ENX4 = SKPTS1 (I, 8)
      ENY4 = SKPTS1 (I, 9)
      ENT4 = SKPTS1 (I, 10)
      ENX5 = SKPTS1 (I+1, 8)
      ENY5 = SKPTS1 (I+1, 9)
      ENT5 = SKPTS1 (I+1, 10)
      CALL PATCH (3, TEST)
      CALL SHKPT
      SKPTS1 (I,1) = X(4)
      SKPTS1 (I,2) = Y(4)
      SKPTS1 (I,3) = T(4)
      SKPTS1 (I,4) = U(4)
      SKPTS1 (I,5) = V(4)
      SKPTS1 (I,6) = P(4)
      SKPTS1 (I,7) = S(4)
      SKPTS1 (I,8) = ENX5
      SKPTS1 (I,9) = ENY5
21    SKPTS1 (I,10) = ENT5
      PRINT 100, ((SKPTS1 (I,J), J=1,10), I=1,MINUS)
100   FORMAT (7E16.8/7X,3E16.8)
C     NOW ALL BODY POINTS ARE CALCULATED
      PRINT 6
6     FORMAT (16H STARTING BDYPTS)
      DO 23 I=1,MINUS
      DO 24 J=1,2
      IJ = I + J - 1
      X(J+2) = BDPTS1(IJ , 1)
      Y(J+2) = BDPTS1(IJ , 2)
      T(J+2) = BDPTS1(IJ , 3)
      U(J+2) = BDPTS1(IJ , 4)
      V(J+2) = BDPTS1(IJ , 5)
      P(J+2) = BDPTS1(IJ , 6)
      S(J+2) = BDPTS1(IJ , 7)
      X(  J  )=FLPTS1(IJ,NPLUS,1)
      Y(  J  )=FLPTS1(IJ,NPLUS,2)
      T(  J  )=FLPTS1(IJ,NPLUS,3)
      U(  J  )=FLPTS1(IJ,NPLUS,4)
      V(  J  )=FLPTS1(IJ,NPLUS,5)
      P(  J  )=FLPTS1(IJ,NPLUS,6)
24    S(  J  )=FLPTS1(IJ,NPLUS,7)
      CALL PATCH (4, TEST)
      CALL BDYPT
      BDPTS1 (I,1) = X(5)
      BDPTS1 (I,2) = Y(5)
      BDPTS1 (I,3) = T(5)
      BDPTS1 (I,4) = U(5)
      BDPTS1 (I,5) = V(5)
      BDPTS1 (I,6) = P(5)
23    BDPTS1 (I,7) = S(5)
      PRINT 4, ((BDPTS1 (I,J), J=1,7), I=1,MINUS)
C     FINALLY ALL FIELD POINTS ARE COMPUTED
      PRINT 7
7     FORMAT (16H STARTING FLDPTS)
      DO 27 I=1,MINUS
      DO 27 J=2,NPLUS
      X(1) = FLPTS1 (I, 1, 1)
      Y(1) = FLPTS1 (I, 1, 2)
      T(1) = FLPTS1 (I, 1, 3)
      U(1) = FLPTS1 (I, 1, 4)
      V(1) = FLPTS1 (I, 1, 5)
      P(1) = FLPTS1 (I, 1, 6)
      S(1) = FLPTS1 (I, 1, 7)
      X(2) = FLPTS1 (I+1, J, 1)
      Y(2) = FLPTS1 (I+1, J, 2)
      T(2) = FLPTS1 (I+1, J, 3)
      U(2) = FLPTS1 (I+1, J, 4)
      V(2) = FLPTS1 (I+1, J, 5)
      P(2) = FLPTS1 (I+1, J, 6)
      S(2) = FLPTS1 (I+1, J, 7)
      X(3) = FLPTS1 (I+1,J-1,1)
      Y(3) = FLPTS1 (I+1,J-1,2)
      T(3) = FLPTS1 (I+1,J-1,3)
      U(3) = FLPTS1 (I+1,J-1,4)
      V(3) = FLPTS1 (I+1,J-1,5)
```

```
      P(3) = FLPTS1 (I+1,J-1,6)
      S(3) = FLPTS1 (I+1,J-1,7)
      CALL PATCH (1, TEST)
      CALL FLOPT
      FLPTS1 (I, 1, 1) = FLPTS1(I, J, 1)
      FLPTS1 (I, 1, 2) = FLPTS1(I, J, 2)
      FLPTS1 (I, 1, 3) = FLPTS1(I, J, 3)
      FLPTS1 (I, 1, 4) = FLPTS1(I, J, 4)
      FLPTS1 (I, 1, 5) = FLPTS1(I, J, 5)
      FLPTS1 (I, 1, 6) = FLPTS1(I, J, 6)
      FLPTS1 (I, 1, 7) = FLPTS1(I, J, 7)
      FLPTS1 (I, J  , 1) = X(4)
      FLPTS1 (I, J  , 2) = Y(4)
      FLPTS1 (I, J  , 3) = T(4)
      FLPTS1 (I, J  , 4) = U(4)
      FLPTS1 (I, J  , 5) = V(4)
      FLPTS1 (I, J  , 6) = P(4)
 27   FLPTS1 (I, J  , 7) = S(4)
      PRINT 4, (((FLPTS1 (I,J,K), K=1,7), J=2,NPLUS), I=1,MINUS)
C     THIS CONCLUDES CALCULATION OF THE EVEN NUMBERED SURFACE.
C     DECISIONS AND INPUT-OUTPUT STATEMENTS ARE HANDLED BELOW
      ITIMS = ITIMS + 1
      M = MINUS
      MINUS = M - 1
      IF (ITIMSR - ITIMS) 29, 29, 61
 61   CONTINUE
C     INTERPOLATION TO SQUARE UP NET NOW FOLLOWS
 93   IF (M - 2) 29, 29, 94
 94   CALL TRUNET (M,N)
      M=M-2
C     EXTRA ROWS OF POINTS ARE ADDED IN THE FOLLOWING STEPS
      INC = INC + 1
      IF (INC1) 990, 991, 990
 991  INC1 = 1
      GO TO (992, 993), INDADD
 990  IF (INADST - INC + 1) 992, 992, 993
 992  CALL ADDROW (KADD, M, N)
      PRINT 996
 996  FORMAT (20H ADDROW DATA FOLLOWS)
      PRINT4,(((FLPTS1(I,J,K),K=1,7),J=2,NPLUS),I=1,M)
      PRINT 100,((SKPTS1(I,K),K=1,10),I=1,M)
      PRINT 4,((ADPTS1(I,K),K=1, 7),I=1,M)
      PRINT 995
 995  FORMAT (19H END OF ADDROW DATA)
      INC = 0
 993  ITIMS = ITIMS + 1
      MINUS=M-1
 60   IF (ITIMSR - ITIMS) 29, 29, 90
 29   READ TAPE 10,NUM
      PRINT 8, NUM
 8    FORMAT (49H THE DATA HAS BEEN WRITTEN ON TAPE 85 AT LOCATION,I3)
      CALL SCAN (NUM)
      WRITE TAPE10,M,N,(((FLPTS1(I,J,K),K=1,7),J=2,NPLUS),I=1,M),
     1(((ADPTS1(I,K),K=1,7),I=1,M),((SKPTS1(I,K),K=1,10),I=1,M)
      END FILE 10
       REWIND 10
      NUM=NUM+1
      WRITE TAPE 10,NUM
      REWIND 10
      CALL EXIT
      END
```

```
*     SUBROUTINE SCAN (N)                                                    SCAN  02
      ENTRY   SCAN                                                           SCAN  03
      LBL     SCAN                                                           SCAN  04
X     TAPENO  B5B             X IS SYNONYMOUS WITH B5, BINARY MODE           SCAN  05
SCAN  SXD     *-2,4           SAVE XR4                                       SCAN  06
      CLA*    1,4             GET COUNT                                      SCAN  07
      TZE     2,4             LEAVE IF N = 0                                 SCAN  08
      STO     N               SAVE IT                                        SCAN  09
      TEFX    *+1             TURN OFF EOF LIGHT                             SCAN  10
START TCOX    *               WAIT UNTIL CHANNEL X READY                     SCAN  11
      STZ     CTR             CLEAR COUNTER                                  SCAN  12
      RTBX                    READ SELECT FOR CHANNEL X                      SCAN  13
      RCHX    READ            RESET AND LOAD CHANNEL X                       SCAN  14
      TCOX    *               WAIT TILL READ DONE                            SCAN  15
      TRCX    PERR            CHECK PARITY ERROR                             SCAN  16
EF    TEFX    EOF             CHECK EOF (TURN OFF INDICATOR)                 SCAN  17
      ETTX                                                                   SCAN  18
      TRA     EOT                                                            SCAN  19
      TRA     START+1         TRY AGAIN                                      SCAN  20
READ  IOCDN   0,0,32767       SKIP 32767 WORDS                               SCAN  21
EOF   CLA     N               PUT IN CURRENT VALUE OF N                      SCAN  22
      LXD     SCAN-2,4                                                       SCAN  23
      SUB     =1B17           SUBTRACT 1 FOR EACH EOF                        SCAN  24
      TZE     2,4             LEAVE IF HAVE N EOF'S                          SCAN  25
      STO     N                                                              SCAN  26
      TRA     START           READ AGAIN                                     SCAN  27
PERR  CLA     CTR             FOR PARITY ERROR                               SCAN  28
      ADD     =1              INCREMENT                                      SCAN  29
      STO     CTR             COUNTER                                        SCAN  30
      SUB     =10             ACCEPT 10 BAD READS                            SCAN  31
      TZE     BAD             IF 10, BAD TAPE                                SCAN  32
      RSRX                                                                   SCAN  33
      CLA     TB5                                                            SCAN  34
      TSX     $(BST),4                                                       SCAN  35
      TRA     START+2         TRY AGAIN, KEEP COUNTER THE SAME               SCAN  36
BAD   TSX     $(SPH),4        PRINT ERROR MESSAGE                            SCAN  37
      PZE     FMT,,*                                                         SCAN  38
      LDQ     N               GIVE CURRENT VALUE OF N                        SCAN  39
      STR                                                                    SCAN  40
      TSX     $(FIL),4                                                       SCAN  41
      CALL    ERROR,MESS                                                     SCAN  42
      TRA     EF              CONTINUE                                       SCAN  43
EOT   CALL    ERROR,TEND                                                     SCAN  44
      CALL    EXIT                                                           SCAN  45
FMT   BCI     5,(21H-BAD TAPE IN SCAN, N=I3)                                 SCAN  46
N     PZE                                                                    SCAN  47
CTR   PZE                                                                    SCAN  48
MESS  BCI     6,0MORE THAN 10 BAD TAPE READS IN SCAN                         SCAN  49
      OCT     777777777777                                                   SCAN  50
TEND  BCI     2,0END OF TAPE                                                 SCAN  51
      OCT     7777777 7777                                                   SCAN  52
TB5   DEC     10B17                                                          SCAN  53
      END                                                                    SCAN  54
```

```
      SUBROUTINE ADDROW (K, MI, NI)
C     ADDROW ADDS K ROWS TO A MI X NI NET BY LINEAR INTERPOLATION SO THAT
C     THE NET BECOMES (MI + K) X NI.  ADDROW ASSUMES THE ROWS ARE EVENLY
C     SPACED AND EVENLY SPACES THE MI + K ROWS BETWEEN THE END POINTS.
C     IF THE ORIGINAL ROWS ARE ONLY SLIGHTLY UNEVENLY SPACED THE RESULTS
C     ARE ONLY SLIGHTLY UNEVEN, BUT IRREGULAR SPACING GIVES IRREGULAR
C     RESULTS.
      COMMON B, Y, T, U, V, A4, UCAP, P, S, ROE4, DXTEST, XTEST, VTEST,
     1BCOE, ENR4, ENV4, ENT4, ENR5, ENV5, ENT5, F, B, SK, TEST, UPLUS, V
     2PLUS, PPLUS, SPLUS
      DIMENSION F(50, 20, 7), B(50, 7), SK(50, 10), X(5), Y(5), T(5),
     1U(5), V(5), P(5), S(5), BCOE(4), FS(50), R(48), AD(96)
      IF (MI + K - 50) 99, 99, 98
 98   PRINT 97
 97   FORMAT (29H M+K EXCEEDS 50 IN ADDROW)
      CALL EXIT
C     FIRST THE BODY POINTS ARE DETERMINED
 99   MIF = MI + K
      S1 = MIF - 1
      S2 = MI - 1
      IDC = 0
      DO 200 I=MI, MIF
200   B(I, 3) = B(I, 3)
 96   DO 1 I=1,MI
 1    FS(I) = B(I, 1)
      N = 2
      M = 2
100   IF (N - MI) 2,  2,910
 2    IDC = IDC + 1
      AM = M - 1
      AN = N - 1
      AD(IDC) = (AM/S1 - AN/S2)
      IF (AD(IDC)) 4, 5, 6
 4    R1 = MI*(N - M     ) + (N - 1)*(K - 1) + M - 1
      R(M - 1) = R1 / S1
      B(M, 1) = R(M-1)*FS(N-1) + (1.- R(M-1))*FS(N)
      M = M + 1
      GO TO 100
 5    B(M, 1) = FS(N)
      M = M + 1
      N = N + 1
      GO TO 100
 6    N = N + 1
      GO TO 100
910   DO 10 J=2,7
      IF (J - 3) 87, 10, 87
 87   IDC = 0
      DO 88 I=1,MI
 88   FS(I) = B(I,J)
      N = 2
      M = 2
103   IF (N - MI) 3, 9, 10
 3    IDC = IDC + 1
C     THE R'S AND AD'S HAVE BEEN CALCULATED ABOVE
      IF ( AD(IDC) ) 43, 53, 63
 43   B(M,J) = R(M - 1)*FS(M - 1) + (1.0 - R(M-1))*FS(N)
      M = M + 1
      GO TO 103
 53   B(M, J) = FS(N)
      M = M + 1
      N = N + 1
      GO TO 103
 63   N = N + 1
      GO TO 103
 10   CONTINUE
C     NEXT THE FIELD POINTS ARE DETERMINED
      NIPLUS = NI + 1
      DO 201 I=MI, MIF
      DO 201 J=2, NIPLUS
201   F(I, J, 3) = F(I, 1, 3)
      DO 20 J=2,NIPLUS
      DO 20 L=1,7
      IF (L - 3) 95, 20, 95
 95   IDC = 0
      DO 11 I=1,MI
 11   FS(I) = F(I, J, L)
      N = 2
      M = 2
101   IF (N - MI) 21, 21, 20
 21   IDC = IDC + 1
      IF (AD(IDC)) 41, 51, 61
 41   F(M, J, L) = R(M-1)*FS(N-1) + (1.- R(M-1))*FS(N)
      M = M + 1
      GO TO 101
 51   F(M, J, L) = FS(N)
      M = M + 1
      N = N + 1
      GO TO 101
 61   N = N + 1
      GO TO 101
 20   CONTINUE
C     FINALLY THE SHOCK POINTS ARE DETERMINED
      DO 202 I=MI, MIF
202   SK (I, 3) = SK (1, 3)
      DO 30 J=1,10
      IF (J - 3) 94, 30, 94
 94   IDC = 0
      DO 12 I=1,MI
 12   FS(I) = SK(I, J)
      N = 2
      M = 2
102   IF (N - MI) 31, 31, 30
 31   IDC = IDC + 1
      IF (AD(IDC)) 42, 52, 62
 42   SK(M, J) = R(M-1)*FS(N-1) +(1.- R(M-1))*FS(N)
      M = M + 1
      GO TO 102
 52   SK(M, J) = FS(N)
      M = M + 1
      N = N + 1
      GO TO 102
 62   N = N + 1
      GO TO 102
 30   CONTINUE
      MI = MIF
      RETURN
      END



      SUBROUTINE TRUNET(M,N)
```

```
C     SUBROUTINE TRUNET SQUARES UP THE NET BOTH IN TIME AND SPACE.
C     SUBROUTINE TRUNET REQUIRES ACCESS TO FUNCTION TINTER WHICH
C     SPECIFIES THE VALUE OF THE CONSTANT TIME SURFACE NEEDED FOR THE
C     INTERPOLATION, AND TO FUNCTIONS THETA AND XINTER.  IT ALSO CALLS THE
C     PATCH, FLDPT, BDYPT AND SHKPT SUBROUTINES.
      DIMENSION  ROWONE(21,7),ROWTWO(21,7),BCOEF(6),X(5),Y(5),T(5),
     1U(5),V(5),P(5),S(5),FLPTS1(50 ,20,7),BDPTS1(90 ,7),SKPTS1(90 ,10)
      COMMON X,Y,T,U,V,A6,UCAP,P,S,ROE6,DXTEST,XTEST,YTEST,BCOEF,ENX6,
     1ENY6,ENT6,ENX5,ENY5,ENT5,FLPTS1,       BDPTS1,SKPTS1 ,TEST
     2,UPLUS,VPLUS,PPLUS,SPLUS
  1   FORMAT(7E16.8)
  2   FORMAT(7E16.8/7(4X,3E16.8))
      NPLUS=N+1
      MIN2=M-2
      MINUS = M -
      NMINUS = N   1
      PRINT 100
 100  FORMAT (16H STARTING TRUNET)
C  TWO ROWS ON THE ODD NUMBERED SURFACE ARE CALCULATED FIRST.
      DO 42 I=1,2
      X(I)=SKPTS1(I,1)
      Y(I)=SKPTS1(I,2)
      T(I)=SKPTS1(I,3)
      U(I)=SKPTS1(I,4)
      V(I)=SKPTS1(I,5)
      P(I)=SKPTS1(I,6)
 42   S(I)=SKPTS1(I,7)
      X(3)=FLPTS1(2,2,1)
      Y(3)=FLPTS1(2,2,2)
      T(3)=FLPTS1(2,2,3)
      U(3)=FLPTS1(2,2,4)
      V(3)=FLPTS1(2,2,5)
      P(3)=FLPTS1(2,2,6)
      S(3)=FLPTS1(2,2,7)
      CALL PATCH (1, TEST)
      CALL FLDPT
      ROWONE(1,1)=X(6)
      ROWONE(1,2)=Y(6)
      ROWONE(1,3)=T(6)
      ROWONE(1,4)=U(6)
      ROWONE(1,5)=V(5)
      ROWONE(1,6)=P(6)
      ROWONE(1,7)=S(6)
      DO 43 J=2,N
      DO 44 K=0,1
      JK=J+K
      X(K+1)=FLPTS1(2,JK,1)
      Y(K+1)=FLPTS1(2,JK,2)
      T(K+1)=FLPTS1(2,JK,3)
      U(K+1)=FLPTS1(2,JK,4)
      V(K+1)=FLPTS1(2,JK,5)
      P(K+1)=FLPTS1(2,JK,6)
 44   S(K+1)=FLPTS1(2,JK,7)
      X(3)=FLPTS1(1,J+1,1)
      Y(3)=FLPTS1(1,J+1,2)
      T(3)=FLPTS1(1,J+1,3)
      U(3)=FLPTS1(1,J+1,4)
      V(3)=FLPTS1(1,J+1,5)
      P(3)=FLPTS1(1,J+1,6)
      S(3)=FLPTS1(1,J+1,7)
      CALL PATCH (1, TEST)
      CALL FLDPT
      ROWONE(J  ,1)=X(6)
      ROWONE(J  ,2)=Y(6)
      ROWONE(J  ,3)=T(6)
      ROWONE(J  ,4)=U(6)
      ROWONE(J  ,5)=V(6)
      ROWONE(J  ,6)=P(6)
 43   ROWONE(J  ,7)=S(6)
      DO 45 K=1, 2
      X(  K)=FLPTS1(  K,NPLUS,1)
      Y(  K)=FLPTS1(  K,NPLUS,2)
      T(  K)=FLPTS1(  K,NPLUS,3)
      U(  K)=FLPTS1(  K,NPLUS,4)
      V(  K)=FLPTS1(  K,NPLUS,5)
      P(  K)=FLPTS1(  K,NPLUS,6)
 45   S(  K)=FLPTS1(  K,NPLUS,7)
      X( 3 ) = BDPTS1(  1, 1)
      Y( 3 ) = BDPTS1(  1, 2)
      T( 3 ) = BDPTS1(  1, 3)
      U( 3 ) = BDPTS1(  1, 4)
      V( 3 ) = BDPTS1(  1, 5)
      P( 3 ) = BDPTS1(  1, 6)
      S( 3 ) = BDPTS1(  1, 7)
      CALL PATCH (1, TEST)
      CALL FLDPT
      ROWONE(N+1,1)=X(6)
      ROWONE(N+1,2)=Y(6)
      ROWONE(N+1,3)=T(6)
      ROWONE(N+1,4)=U(6)
      ROWONE(N+1,5)=V(6)
      ROWONE(N+1,6)=P(6)
 90   ROWONE(N+1,7)=S(6)
      TSPEC=TINTER(ROWONE)
      PRINT 1,((ROWONE(I,J),J=1,7),I=1,NPLUS)
      DO 94 I=2,MINUS
      DO 46 J=1,2
      IJ=I+J-1
      X(J)=SKPTS1(IJ,1)
      Y(J)=SKPTS1(IJ,2)
      T(J)=SKPTS1(IJ,3)
      U(J)=SKPTS1(IJ,4)
      V(J)=SKPTS1(IJ,5)
      P(J)=SKPTS1(IJ,6)
 46   S(J)=SKPTS1(IJ,7)
      X(3)=FLPTS1(I , 2, 1)
      Y(3)=FLPTS1(I , 2, 2)
      T(3)=FLPTS1(I , 2, 3)
      U(3)=FLPTS1(I , 2, 4)
      V(3)=FLPTS1(I , 2, 5)
      P(3)=FLPTS1(I , 2, 6)
      S(3)=FLPTS1(I , 2, 7)
      CALL PATCH (1, TEST)
      CALL FLDPT
      ROWTWO(1,1)=X(6)
      ROWTWO(1,2)=Y(6)
      ROWTWO(1,3)=T(6)
      ROWTWO(1,4)=U(6)
      ROWTWO(1,5)=V(6)
      ROWTWO(1,6)=P(6)
      ROWTWO(1,7)=S(6)
      DO 47 J=2,N
```

```
      DO 48 K=1,2
      IK = I + K - 1
      X(K  )=FLPTS1(IK,J,1)
      Y(K  )=FLPTS1(IK,J,2)
      T(K  )=FLPTS1(IK,J,3)
      U(K  )=FLPTS1(IK,J,4)
      V(K  )=FLPTS1(IK,J,5)
      P(K  )=FLPTS1(IK,J,6)
   48 S(K  )=FLPTS1(IK,J,7)
      X(3)=FLPTS1(I+1,J+1,1)
      Y(3)=FLPTS1(I+1,J+1,2)
      T(3)=FLPTS1(I+1,J+1,3)
      U(3)=FLPTS1(I+1,J+1,4)
      V(3)=FLPTS1(I+1,J+1,5)
      P(3)=FLPTS1(I+1,J+1,6)
      S(3)=FLPTS1(I+1,J+1,7)
      CALL PATCH (1, TEST)
      CALL FLDPT
      ROWTWO(J  ,1)=X(4)
      ROWTWO(J  ,2)=Y(4)
      ROWTWO(J  ,3)=T(4)
      ROWTWO(J  ,4)=U(4)
      ROWTWO(J  ,5)=V(4)
      ROWTWO(J  ,6)=P(4)
   47 ROWTWO(J  ,7)=S(4)
      DO 49 K=1,2
      IK = I + K - 1
      X(  K)=FLPTS1(IK,NPLUS,1)
      Y(  K)=FLPTS1(IK,NPLUS,2)
      T(  K)=FLPTS1(IK,NPLUS,3)
      U(  K)=FLPTS1(IK,NPLUS,4)
      V(  K)=FLPTS1(IK,NPLUS,5)
      P(  K)=FLPTS1(IK,NPLUS,6)
   49 S(  K)=FLPTS1(IK,NPLUS,7)
      X( 3 ) = BDPTS1(I , 1)
      Y( 3 ) = BDPTS1(I , 2)
      T( 3 ) = BDPTS1(I , 3)
      U( 3 ) = BDPTS1(I , 4)
      V( 3 ) = BDPTS1(I , 5)
      P( 3 ) = BDPTS1(I , 6)
      S( 3 ) = BDPTS1(I , 7)
      CALL PATCH (1, TEST)
      CALL FLDPT
      ROWTWO(N+1,1)=X(4)
      ROWTWO(N+1,2)=Y(4)
      ROWTWO(N+1,3)=T(4)
      ROWTWO(N+1,4)=U(4)
      ROWTWO(N+1,5)=V(4)
      ROWTWO(N+1,6)=P(4)
   51 ROWTWO(N+1,7)=S(4)
      PRINT 1,((ROWTWO(K,J),J=1,7),K=1,NPLUS)
C  THE TWO AUXILIARY ROWS ON THE ODD NUMBERED SURFACE HAVE BEEN COMPUTED AND
C  THE FIRST ROW OF THE EVEN NUMBERED SURFACE MAY NOW BE CALCULATED.
      DO 90 J=1,2
      K2=I+J-1
      X(J+2) = SKPTS1 (K2, 1)
      Y(J+2) = SKPTS1 (K2, 2)
      T(J+2) = SKPTS1 (K2, 3)
      U(J+2) = SKPTS1 (K2, 4)
      V(J+2) = SKPTS1 (K2, 5)
      P(J+2) = SKPTS1 (K2, 6)
   90 S(J+2) = SKPTS1 (K2, 7)
      ENX4 = SKPTS1 (I+1, 8)
      ENY4 = SKPTS1 (I+1, 9)
      ENT4 = SKPTS1 (I+1, 10)
      ENX5 = SKPTS1 (I, 8)
      ENY5 = SKPTS1 (I, 9)
      ENT5 = SKPTS1 (I, 10)
      X(1) = ROWTWO (1,1)
      Y(1) = ROWTWO (1,2)
      T(1) = ROWTWO (1,3)
      U(1) = ROWTWO (1,4)
      V(1) = ROWTWO (1,5)
      P(1) = ROWTWO (1,6)
      S(1) = ROWTWO (1,7)
      X(2)=ROWONE(1,1)
      Y(2)=ROWONE(1,2)
      T(2)=ROWONE(1,3)
      U(2)=ROWONE(1,4)
      V(2)=ROWONE(1,5)
      P(2)=ROWONE(1,6)
      S(2)=ROWONE(1,7)
      CALL PATCH (3, TEST)
      CALL SHKPT
      CT=SQRTF((X(4)-SKPTS1(I  ,1))**2+(Y(4)-SKPTS1(I  ,2))**2+
     1(T(4)-SKPTS1(I  ,3))**2)
      SKPTS1(I-1,1)=(X(4)-SKPTS1(I  ,1))*(TSPEC-SKPTS1(I  ,3))/
     1(T(4)-SKPTS1(I  ,3)) +SKPTS1(I  ,1)
      SKPTS1(I-1,2)=(Y(4)-SKPTS1(I  ,2))*(TSPEC-SKPTS1(I  ,3))/
     1(T(4)-SKPTS1(I  ,3)) + SKPTS1(I  ,2)
      SKPTS1(I-1,3)=TSPEC
      CP=SQRTF((SKPTS1(I-1,1)-X(4))**2+(SKPTS1(I-1,2)-Y(4))**2+
     1(SKPTS1(I-1,3)-T(4))**2)
      SKPTS1(I-1,4)=U(4)-CP*(U(4)-SKPTS1(I  ,4))/CT
      SKPTS1(I-1,5)=V(4)-CP*(V(4)-SKPTS1(I  ,5))/CT
      SKPTS1(I-1,6)=P(4)-CP*(P(4)-SKPTS1(I  ,6))/CT
      SKPTS1(I-1,7)=S(4)-CP*(S(4)-SKPTS1(I  ,7))/CT
      SKPTS1(I-1,8)=ENX5-CP*(ENX5-SKPTS1(I  ,8))/CT
      SKPTS1(I-1,9)=ENY5-CP*(ENY5-SKPTS1(I  ,9))/CT
   92 SKPTS1(I-1,10)=ENT5-CP*(ENT5-SKPTS1(I  ,10))/CT
C     NEW FIELD POINTS ALONG ONE ROW  ARE CALCULATED NOW
      DO 91 J=1,N
      DO 92 K=1,2
      JK = J + K - 1
      X(  K)=ROWONE(JK,1)
      Y(  K)=ROWONE(JK,2)
      T(  K)=ROWONE(JK,3)
      U(  K)=ROWONE(JK,4)
      V(  K)=ROWONE(JK,5)
      P(  K)=ROWONE(JK,6)
   92 S(  K)=ROWONE(JK,7)
      X( 3 ) = ROWTWO(J+1,1)
      Y( 3 ) = ROWTWO(J+1,2)
      T( 3 ) = ROWTWO(J+1,3)
      U( 3 ) = ROWTWO(J+1,4)
      V( 3 ) = ROWTWO(J+1,5)
      P( 3 ) = ROWTWO(J+1,6)
      S( 3 ) = ROWTWO(J+1,7)
      CALL PATCH (1, TEST)
      CALL FLDPT
      L = J + 1
      CT=SQRTF((X(4)-FLPTS1(I  ,L,1))**2+(Y(4)-FLPTS1(I  ,L,2))**2
```

```
     1+(T(6)-FLPTS1(I  ,L,3))**2)
      FLPTS1(I-1,J,1)=(X(6)-FLPTS1(I  ,L,1))*(TSPEC-FLPTS1(I  ,L,3))/
     1(T(6)-FLPTS1(I  ,L,3))+FLPTS1(I  ,L,1)
      FLPTS1(I-1,J,2)=(Y(6)-FLPTS1(I  ,L,2))*(TSPEC-FLPTS1(I  ,L,3))/
     1(T(6)-FLPTS1(I  ,L,3))+FLPTS1(I  ,L,2)
      FLPTS1(I-1,J,3)=TSPEC
      CP=SQRTF((FLPTS1(I-1,J,1)-X(6))**2+(FLPTS1(I-1,J,2)-Y(6))**2
     1+(FLPTS1(I-1,J,3)-T(6))**2)
      FLPTS1(I-1,J,4)=U(6)-CP*(U(6)-FLPTS1(I  ,L,4))/CT
      FLPTS1(I-1,J,5)=V(6)-CP*(V(6)-FLPTS1(I  ,L,5))/CT
   33 FLPTS1(I-1,J,6)=P(6)-CP*(P(6)-FLPTS1(I  ,L,6))/CT
   51 FLPTS1(I-1,J,7)=S(6)-CP*(S(6)-FLPTS1(I  ,L,7))/CT
C     NEW BODY POINTS WILL NOW BE CALCULATED
      DO 93 J=1,2
      IJ=I+J-1
      X(J+2) = BDPTS1(IJ, 1)
      Y(J+2) = BDPTS1(IJ, 2)
      T(J+2) = BDPTS1(IJ, 3)
      U(J+2) = BDPTS1(IJ, 4)
      V(J+2) = BDPTS1(IJ, 5)
      P(J+2) = BDPTS1(IJ, 6)
   93 S(J+2) = BDPTS1(IJ, 7)
      X(1)=ROWONE(N+1,1)
      Y(1)=ROWONE(N+1,2)
      T(1)=ROWONE(N+1,3)
      U(1)=ROWONE(N+1,4)
      V(1)=ROWONE(N+1,5)
      P(1)=ROWONE(N+1,6)
      S(1)=ROWONE(N+1,7)
      X(2)=ROWTWO(N+1,1)
      Y(2)=ROWTWO(N+1,2)
      T(2)=ROWTWO(N+1,3)
      U(2)=ROWTWO(N+1,4)
      V(2)=ROWTWO(N+1,5)
      P(2)=ROWTWO(N+1,6)
      S(2)=ROWTWO(N+1,7)
      CALL PATCH (4, TEST)
      CALL BDYPT
      CT=SQRTF((X(5)-BDPTS1(I  ,1))**2+(Y(5)-BDPTS1(I  ,2))**2
     1+(T(5)-BDPTS1(I  ,3))**2)
      BDPTS1(I-1,1)=(X(5)-BDPTS1(I  ,1))*(TSPEC-BDPTS1(I  ,3))/
     1(T(5)-BDPTS1(I  ,3))+BDPTS1(I  ,1)
      BDPTS1(I-1,2)=(Y(5)-BDPTS1(I  ,2))*(TSPEC-BDPTS1(I  ,3))/
     1(T(5)-BDPTS1(I  ,3))+BDPTS1(I  ,2)
      BDPTS1(I-1,3)=TSPEC
      CP=SQRTF((BDPTS1(I-1,1)-X(5))**2+(BDPTS1(I-1,2)-Y(5))**2
     1+(BDPTS1(I-1,3)-T(5))**2)
      BDPTS1(I-1,4)=U(5)-CP*(U(5)-BDPTS1(I  ,4))/CT
      BDPTS1(I-1,5)=V(5)-CP*(V(5)-BDPTS1(I  ,5))/CT
      BDPTS1(I-1,6)=P(5)-CP*(P(5)-BDPTS1(I  ,6))/CT
   34 BDPTS1(I-1,7)=S(5)-CP*(S(5)-BDPTS1(I  ,7))/CT
      DO 94 J=1,NPLUS
      DO 94 K=1,7
   94 ROWONE(J ,K)=ROWTWO(J ,K)
      PRINT 101
  101 FORMAT (23H STARTING FLDPTS TRUNET)
      PRINT 1,(((FLPTS1(I,J,K),K=1,7),J=1,N),I=1,MIN2)
      PRINT 102
  102 FORMAT (23H STARTING BDYPTS TRUNET)
      PRINT 1, ((BDPTS1 (I,K), K=1,7), I=1,MIN2)
      PRINT 2,((SKPTS1(I,K),K=1,10),I=1,MIN2)
C  THE MESH HAS NOW BEEN SQUARED UP IN TIME. SQUARING IN SPACE IS ACCOMPLISHED
C  BELOW.
      DO 68 L=1,5
   68 T(L)=TSPEC
      DO 67 I=1,MIN2
      CP = SKPTS1(I,2) - BDPTS1(I,2)
      CT = SKPTS1(I,1) - BDPTS1(I,1)
      PHI = THETA (CT, CP)
      SINPHI = SINF(PHI)
      COSPHI = COSF(PHI)
      ROWTWO(1,1) = CT*COSPHI + CP*SINPHI
      ROWTWO (1, 3) = NPLUS
      DO 62 J=1,N
      ROWTWO(1,2) = N - J + 1
      ROWONE(J,1) = ROWTWO(1,2)*ROWTWO(1,1)*COSPHI/ROWTWO (1,3) +
     1 BDPTS1(I,1)
   62 ROWONE(J,2) = ROWTWO(1,2)*ROWTWO(1,1)*SINPHI/ROWTWO (1,3) +
     1 BDPTS1(I,2)
C  THE DEPENDENT VARIABLES ARE NOW OBTAINED BY INTERPOLATION.
   61 IF (I - MIN2) 80, 81, 81
   80 DO 69 J=1,NMINUS
      DO 64 K=1,2
      JK=J+K-1
      X(K)=FLPTS1(I,JK,1)
      Y(K)=FLPTS1(I,JK,2)
      U(K)=FLPTS1(I,JK,4)
      V(K)=FLPTS1(I,JK,5)
      P(K)=FLPTS1(I,JK,6)
   64 S(K)=FLPTS1(I,JK,7)
      X(3)=FLPTS1(I+1,J,1)
      Y(3)=FLPTS1(I+1,J,2)
      U(3)=FLPTS1(I+1,J,4)
      V(3)=FLPTS1(I+1,J,5)
      P(3)=FLPTS1(I+1,J,6)
      S(3)=FLPTS1(I+1,J,7)
      X(5)=ROWONE(J,1)
      Y(5)=ROWONE(J,2)
      CALL XINTER (X, Y, T, U, V, P, S)
      FLPTS1(I,J,4)=U(5)
      FLPTS1(I,J,5)=V(5)
      FLPTS1(I,J,6)=P(5)
      FLPTS1(I,J,7)=S(5)
   35 DO 65 L=1,2
   65 FLPTS1(I,J,L)= OWONE(J,L)
      DO 66 K=1,2
      IK=I+K-1
      X(K)=FLPTS1(IK,N,1)
      Y(K)=FLPTS1(IK,N,2)
      U(K)=FLPTS1(IK,N,4)
      V(K)=FLPTS1(IK,N,5)
      P(K)=FLPTS1(IK,N,6)
   66 S(K)=FLPTS1(IK,N,7)
      X(3)=BDPTS1(I,1)
      Y(3)=BDPTS1(I,2)
      U(3)=BDPTS1(I,4)
      V(3)=BDPTS1(I,5)
      P(3)=BDPTS1(I,6)
      S(3)=BDPTS1(I,7)
      X(5)=ROWONE(N,1)
      Y(5)=ROWONE(N,2)
      CALL XINTER (X, Y, T, U, V, P, S)
```

```
      FLPTS1(I,N,4)=U(5)
      FLPTS1(I,N,5)=V(5)
      FLPTS1(I,N,6)=P(5)
      FLPTS1(I,N,7)=S(5)
      DO 67 L=1,2
   67 FLPTS1(I,N,L)=ROWONE(N,L)
   81 DO 72 J=1,MINUS
      DO 71 K=1,2
      JK=J+K-1
      X(K)=FLPTS1(N-2,JK,1)
      Y(K)=FLPTS1(N-2,JK,2)
      U(K)=FLPTS1(N-2,JK,4)
      V(K)=FLPTS1(N-2,JK,5)
      P(K)=FLPTS1(N-2,JK,6)
   71 S(K)=FLPTS1(N-2,JK,7)
      X(3)=FLPTS1(N-3,J,1)
      Y(3)=FLPTS1(N-3,J,2)
      U(3)=FLPTS1(N-3,J,4)
      V(3)=FLPTS1(N-3,J,5)
      P(3)=FLPTS1(N-3,J,6)
      S(3)=FLPTS1(N-3,J,7)
      X(4)=ROWONE(J,1)
      Y(4)=ROWONE(J,2)
      CALL XINTER (X, Y, T, U, V, P, S)
      FLPTS1(N-2,J,4)=U(5)
      FLPTS1(N-2,J,5)=V(5)
      FLPTS1(N-2,J,6)=P(5)
   97 FLPTS1(N-2,J,7)=S(5)
      DO 72 L=1,2
   72 FLPTS1(N-2,J,L)=ROWONE(J,L)
      DO 73 K=1,2
      IK=M-1-K
      X(K)=BDPTS1(IK,1)
      Y(K)=BDPTS1(IK,2)
      U(K)=BDPTS1(IK,4)
      V(K)=BDPTS1(IK,5)
      P(K)=BDPTS1(IK,6)
   73 S(K)=BDPTS1(IK,7)
      X(3)=FLPTS1(N-2,M,1)
      Y(3)=FLPTS1(N-2,M,2)
      U(3)=FLPTS1(N-2,M,4)
      V(3)=FLPTS1(N-2,M,5)
      P(3)=FLPTS1(N-2,M,6)
      S(3)=FLPTS1(N-2,M,7)
      X(4)=ROWONE(M,1)
      Y(4)=ROWONE(M,2)
      CALL XINTER (X, Y, T, U, V, P, S)
      FLPTS1(N-2,M,4)=U(5)
      FLPTS1(N-2,M,5)=V(5)
      FLPTS1(N-2,M,6)=P(5)
      FLPTS1(N-2,M,7)=S(5)
      DO 74 L=1,2
   74 FLPTS1(N-2,M,L)=ROWONE(M,L)
      PRINT 1,(((FLPTS1(I,J,K),K=1,7),J=1,M),I=1,MIN2)
      PRINT 103
  103 FORMAT (12H BACK TO EXC)
      DO 39 I=1,MIN2
      DO 39 J=1,M
       DO 39 L=1,7
      K=M-J+1
   39  FLPTS1(I,K+1,L)=FLPTS1(I,K,L)
C  THE ENTIRE FIELD HAS NOW BEEN SQUARED UP BOTH IN TIME AND SPACE.
      RETURN
      END



      FUNCTION TINTER(ROWONE)
C  THIS SUBROUTINE DETERMINES THE TIME AT WHICH TRUNET LOCATES THE CONSTANT TIME
C  PLANE.
      DIMENSION F(50, 20, 7), B(50, 7), SK(50, 10), X(5), Y(5), T(5),
     1 U(5), V(5), P(5), S(5), BCOE(4), ROWONE(21, 7)
      COMMON X, Y, T, U, V, A4, UCAP, P, S, ROE4, DXTEST, XTEST, VTEST,
     1BCOE, ENX4, ENY4, ENT4, ENX5, ENY5, ENT5, F, B, SK, TEST, UPLUS,
     2VPLUS, PPLUS, SPLUS
      DO 10 I=1,2
      X(I) = F(I+19, 8, 1)
      Y(I) = F(I+19, 8, 2)
      T(I) = F(I+19, 8, 3)
      U(I) = F(I+19, 8, 4)
      V(I) = F(I+19, 8, 5)
      P(I) = F(I+19, 8, 6)
      S(I) = F(I+19, 8, 7)
      X(I+2) = B(I+19, 1)
      Y(I+2) = B(I+19, 2)
      T(I+2) = B(I+19, 3)
      U(I+2) = B(I+19, 4)
      V(I+2) = B(I+19, 5)
      P(I+2) = B(I+19, 6)
   10 S(I+2) = B(I+19, 7)
      CALL PATCH (4, TEST)
      CALL BDYPT
      TINTER = T(5)
      RETURN
      END



      SUBROUTINE PATCH (N, TEST)
C     N = 1 FOR 3 POINT INPUT WITH CIRCLE FIT, N = 2 FOR 4 POINT INPUT
C     WITH CIRCLE FIT, N = 3 FOR SHKPT TYPE AND N = 4 FOR BDYPT TYPE
C     SIMPLE INTERPOLATIONS WITH 4 POINT INPUT, TEST = CONVERGENCE
C     CRITERION.  THIS SUBROUTINE CALLS THE FUNCTION THETA.
      COMMON X, Y, T, U, V, A4, UCAP, P, S, ROE4, DXTEST, XTEST, VTEST,
     1BCOEF, ENX4, ENY4, ENT4, ENX5, ENY5, ENT5
      DIMENSION X(5), Y(5), T(5), U(5), V(5), P(5), S(5), BCOEF(4), XX(4
     1), YY(4), XA(3), YA(3), A(7,7), B(7), C(7)
C     FIRST THE COORDINATES ARE TRANSFORMED
      GO TO (700, 700, 800, 900), N
  700 C(1) = X(2) - X(1)
      C(2)=T(2)-T(1)
      THE=THETA(C(1),C(2))
      SINTHE=SINF(THE)
      COSTHE =COSF(THE)
      C(1)=(X(2)-X(1))*COSTHE+(T(2)-T(1))*SINTHE
      C(2)=Y(2)-Y(1)
      PHI=THETA(C(1),C(2))
      SINPHI=SINF(PHI)
      COSPHI=COSF(PHI)
      C(1)= (Y(3)-Y(1))*COSPHI-((X(3)-X(1))*COSTHE+(T(3)-T(1)))*
```

```
     1SINTHE)*SINPHI
      C(2)=(T(3)-T(1))*COSTHE-(X(3)-X(1))*SINTHE
      PSI=THETA(C(1),C(2))
      SINPSI = SINF (PSI)
      COSPSI = COSF (PSI)
      XX(1) = 0.0
      DO 10 I=2,3
 10   XX(I) = ((X(I) - X(1))*COSTHE + (T(I) - T(1))*SINTHE) * COSPHI + (
     1 Y(I) - Y(1))*SINPHI
      GO TO (11, 12), N
 11   XX(4) = 0.0
      GO TO 13
 12   XX(4) = ((X(4) - X(1))*COSTHE + (T(4) - T(1))*SINTHE)*COSPHI + (Y(
     14) - Y(1))*SINPHI
 13   YY(1) = 0.0
      YY(2) = 0.0
      YY(3) = ((Y(3) - Y(1))*COSPHI - ((X(3) - X(1))*COSTHE + (T(3) - T(
     11))*SINTHE)*SINPHI)*COSPSI+ ((T(3) - T(1))*COSTHE - (X(3) - X(1))*
     2SINTHE)*SINPSI
      GO TO (14, 15), N
 14   YY(4) = 0.0
      GO TO 16
 15   YY(4) = ((Y(4) - Y(1))*COSPHI - ((X(4) - X(1))*COSTHE + (T(4) - T(
     11))*SINTHE)*SINPHI)*COSPSI+ ((T(4) - T(1))*COSTHE - (X(4) - X(1))*
     2SINTHE)*SINPSI
C     NEXT THE FIRST ESTIMATES ARE CALCULATED
   16 DO 17 I=2,3
 17   XA(I) = (XX(I) + XX(I+1))/2.0
      YA(2) = YY(3) / 2.0
      YA(3) = (YY(3) + YY(4))/2.0
      R2=  ((ABSF(XX(2)))  **SQRTF ((XX(2)-XX(3))**2  +YY(3)**2) + SQRTF
     1((  XX(3)-XX(4))**2+(YY(3)-YY(4))**2 )+ SQRTF( XX(4)**2+YY(4)**2
     2))  /8.0 )**2
      EN = N
      XZERO =-(XX(2) + XX(3) + XX(4))/(2.0 * EN)
      YZERO =-(YY(3) + YY(4))/(2.0 * EN)
      XA(1)=-XZERO
      YA(1)=0.0
C     THE MATRIX EQUATION IS SET UP IN THE FOLLOWING STEPS
      K = 0
 20   DO 18 I=1,7
      DO 18 J=1,7
      A(I,J) = 0.0
 18   B(I) = 0.0
      DO 21 I=1,2
      A(I,2*I-1)=YY(I+2)-YY(I+1)
      A(I,2*I)=XX(I+1)-XX(I+2)
      A(3,6*I-3)=A(1,2)
      A(3,6*I-2)=-A(1,1)
      A(6,6*I-3)=2.0*(XA(2)+XZERO)
      A(6,6*I-2)=2.0*(YA(2)+YZERO)
      A(7,2*I+1)=2.0*(XA(3)+XZERO)
      A(7,2*I+2)=2.0*(YA(3)+YZERO)
      B(I)=(YY(I+1)-YY(I+2))*XA(I+1)+(XX(I+2)-XX(I+1))*YA(I+1)+XX(I+1)
     1*YY(I+2)-XX(I+2)*YY(I+1)
      B(I+2)=(XX(I+2)-XX(I+1))*XA(I+1)+(YY(I+2)-YY(I+1))*YA(I+1)
     1+(XX(I+2)-XX(I+1))*XZERO+(YY(I+2)-YY(I+1))*YZERO
   21 B(I+5)=-(XA(I+1)+XZERO)**2-(YA(I+1)+YZERO)**2+R2
      DO 19 I=1,2
      A(6,2*I+1)=A(2,6)
   19 A(6,2*I+2)=-A(2,5)
      DO 3 I=5,7
    3 A(I,7)=-1.0
      A(5,6)=2.0*YZERO
      B(9)=R2-YZERO**2
      K = K + 1
      IF (K - 50) 100, 100, 101
  101 PRINT 102
  102 FORMAT (21H EXCESS LOOPING PATCH)
      RETURN
C     NOW SOLVE THE EQUATIONS
  100  DET = 1.0
      M = RSIMEQF(7, 7, 1, A, B, DET, C)
      GO TO (22, 93, 94), M
   93 PRINT 1
      CALL EXIT
   94 PRINT 2
      CALL EXIT
    1 FORMAT (21H OVER/UNDERFLOW PATCH)
    2 FORMAT (22H MATRIX SINGULAR PATCH)
C     THE TEST STATEMENTS FOR THE ITERATION FOLLOW
   22 IF(XA(2))60,25,60
   60 IF(ABSF(A(1,1))-TEST*ABSF(XA(2)))25,31,31
   25 IF(YA(2))61,26,61
   61 IF(ABSF(A(2,1))-TEST*ABSF(YA(2)))26,31,31
   26 IF(XA(3))62,27,62
   62 IF(ABSF(A(3,1))-TEST*ABSF(XA(3)))27,31,31
   27 IF(YA(3))63,28,63
   63 IF(ABSF(A(4,1))-TEST*ABSF(YA(3)))28,31,31
   28 IF(XZERO)64,29,64
   64 IF(ABSF(A(5,1))-TEST*ABSF(XZERO))29,31,31
   29 IF(YZERO)65,30,65
   65 IF(ABSF(A(6,1))-TEST*ABSF(YZERO))30,31,31
   30 IF(R2)66,33,66
   66 IF(ABSF(A(7,1))-TEST*ABSF(R2))33,31,31
   33 IDUP=1
      GO TO 23
   31 IDUP=2
   23 XA(2)=XA(2)+A(1,1)
      YA(2)=YA(2)+A(2,1)
      XA(3)=XA(3)+A(3,1)
      YA(3)=YA(3)+A(4,1)
      XZERO=XZERO+A(5,1)
      YZERO=YZERO+A(6,1)
      R2=R2+A(7,1)
      GO TO(4,20), IDUP
C     INTERPOLATION FOLLOWS
    4 YA(1)=0.0
      XA(1)=-XZERO
      DO 34 I=1,3
   34 C(I) = SQRTF(((XA(I) - XX(I))**2 + (YA(I) - YY(I))**2)/ ((XX(I+1)
     1 - XX(I))**2 + (YY(I+1) - YY(I))**2))
      GO TO (35, 36), N
   35 DO 37 I=1,2
      U(I) = (U(I+1) - U(I))*C(I) + U(I)
      V(I) = (V(I+1) - V(I))*C(I) + V(I)
      P(I) = (P(I+1) - P(I))*C(I) + P(I)
   37 S(I) = (S(I+1) - S(I))*C(I) + S(I)
      U(3)=(U(1)-U(3))*C(3)+U(3)
      V(3)=(V(1)-V(3))*C(3)+V(3)
      P(3)=(P(1)-P(3))*C(3)+P(3)
      S(3)=(S(1)-S(3))*C(3)+S(3)
```

```
      GO TO 50
   36 DO 38 I=1,2
      J = 2*I -1
      U(I) = (U(J+1) - U(J))*C(J) + U(J)
      V(I) = (V(J+1) - V(J))*C(J) + V(J)
      P(I) = (P(J+1) - P(J))*C(J) + P(J)
   38 S(I) = (S(J+1) - S(J))*C(J) + S(J)
C     FINALLY, TRANSFORM BACK TO THE ORIGINAL COORDINATES
   50 C(1) = X(1)
      C(2) = Y(1)
      C(3) = T(1)
      GO TO (39, 40), N
   39 N = 3
      GO TO 41
   40 N = 2
   41 DO 42 I=1,N
      GO TO (43, 44) , N
   43 J = 1
      GO TO 45
   44 J = 2*I-1
   45 X(I) = (XA(J) *COSPHI - YA(J)   *COSPSI*SINPHI)*COSTHE - YA(J)   *
     1 SINPSI*SINTHE + C(1)
      Y(I) = YA(J)   *COSPSI*COSPHI + XA(J) *SINPHI + C(2)
   42 T(I) = YA(J)   *SINPSI*COSTHE + (XA(J) *COSPHI - YA(J)   *COSPSI*S
     1INPHI)*SINTHE + C(3)
      IF(T(1)-0.0000001)70,70,71
   70 T(1)=0.0
   71 IF(T(2)-0.0000001)72,72,73
   72 T(2)=0.0
   73 IF(T(3)-0.0000001)74,74,5
   74 T(3)=0.0
    5 RETURN
C     THE SHIPT TYPE SIMPLE INTERPOLATION FOLLOWS
  800 X(3) = (X(3) + X(4))/2.0
      Y(3) = (Y(3) + Y(4))/2.0
      T(3) = (T(3) + T(4))/2.0
      U(3) = (U(3) + U(4))/2.0
      V(3) = (V(3) + V(4))/2.0
      P(3) = (P(3) + P(4))/2.0
      S(3) = (S(3) + S(4))/2.0
      ENX4 = (ENX4 + ENX5)/2.0
      ENY4 = (ENY4 + ENY5)/2.0
      ENT4 = (ENT4 + ENT5)/2.0
      DO 801 I=1,2
      X(I) = 0.4*X(I) + 0.6*X(3)
      Y(I) = 0.4*Y(I) + 0.6*Y(3)
      T(I) = 0.4*T(I) + 0.6*T(3)
      U(I) = 0.4*U(I) + 0.6*U(3)
      V(I) = 0.4*V(I) + 0.6*V(3)
      P(I) = 0.4*P(I) + 0.6*P(3)
  801 S(I) = 0.4*S(I) + 0.6*S(3)
      RETURN
C     THE BOYPT TYPE SIMPLE IN'<MPOLATION FOLLOWS
  900 X(5) = (X(3) + X(4))/2.0
      Y(5) = (Y(3) + Y(4))/2.0
      T(5) = (T(3) + T(4))/2.0
      U(5) = (U(3) + U(4))/2.0
      V(5) = (V(3) + V(4))/2.0
      P(5) = (P(3) + P(4))/2.0
      S(5) = (S(3) + S(4))/2.0
      DO 901 I=1,2
      X(I) = 0.4*X(I) + 0.6*X(5)
      Y(I) = 0.4*Y(I) + 0.6*Y(5)
      T(I) = 0.4*T(I) + 0.6*T(5)
      U(I) = 0.4*U(I) + 0.6*U(5)
      V(I) = 0.4*V(I) + 0.6*V(5)
      P(I) = 0.4*P(I) + 0.6*P(5)
  901 S(I) = 0.4*S(I) + 0.6*S(5)
      RETURN
      END
```

```
      SUBROUTINE FLOPT
C  THIS SUBROUTINE CALLS SUBROUTINE XINTER AND FUNCTIONS ADET AND ROEDET.
      DIMENSION AA(6), BCOEF(6), RESIDU(3), GAMMA(6), P(5), ROEAVE(3),
     1ROEA(6), UAVE(3), VAVE(3), AAVE(3), X(5), Y(5), T(5), U(5), V(5),
     2S(5), PAVE(3), SAVE(3), A(3), B(3),
     3C(3), D(3), UABC(3), E(3), F(3), G(3), H(3), FX(3), FY(3),
     4FZ(3), AARRAY(3,3), BCOL(3   ), DDBETA(3,3), VDELBE(3,3), DEL(3),
     5EL(3), EM(3), EK1(3), EK2(3), EK3(3), BETA(3,3,3), BARRAY(3,3)
      COMMON X, Y, T, U, V, AA, UCAP, P, S, ROE4, DXTEST, XTEST, VTEST,
     1BCOEF, ENX4, ENY4, ENT4, ENX5, ENY5, ENT5
      EQUIVALENCE (C(1), DEL(1)), (UABC(1), EL(1)), (G(1), EM(1), GAMMA(
     11)), (FX(1), EK1(1), GAMMA(4)), (FY(1), EK2(1)), (FZ(1), EK3(1)),
     2(AC, UAVE5), (BC, DDBETA(1)), VAVE5), (CC, DDBETA(2), BAG), (DC, DD
     3BETA(3)), (DEN, DDBETA(4)), (XC, DDBETA(5)), (YC, DDBETA(6)), (ZC,
     4 DDBETA(7)), (AOAVG, DDBETA(8)), (UAVG, DDBETA(9)), (VAVG, VDELBE(
     51)), (DELTAX, VDELBE(3)), (DELTAY, VDELBE(4)), (DELTAZ, VDELBE(5))
     6,(DET, VDELBE(6)), (VTOT, VDELBE(7))
C  PRELIMINARY CALCULATIONS OF AVERAGES START HERE.
      DO 10 I=1,3
      UAVE(I) = U(I)
      VAVE(I) = V(I)
      ROEA(I) = ROEDET (P(I), S(I))
      AA(I) = ADET (P(I), S(I))
      AAVE(I) = AA(I)
   10 ROEAVE(I) = ROEA(I)
      U(5) = (U(1) + U(2) + U(3))/3.0
      V(5) = (V(1) + V(2) + V(3))/3.0
      U(4) = U(5)
      V(4) = V(5)
      PX4 = (P(1) + P(2) + P(3))/3.0
      VTOTX=SQRTF((U(1) + U(2) + U(3))**2 + (V(1) + V(2) + V(3))**2)/3.0
C  CALCULATION OF THE FIRST ESTIMATE OF THE COORDINATES FOLLOWS.
      DO 11 I=1,2
      A(I)= X(I+1)- X(I)
      B(I)= Y(I+1) -Y(I)
   11 C(I) = T(I+1) - T(I)
      AC= B(1)*C(2) - B(2)*C(1)
      BC= C(1)*A(2) - C(2)*A(1)
      CC= A(1)*B(2) - A(2)*B(1)
      DO 12 I=1,2
   12   D(I)= A(I)*( X(I) + X(I+1))/2.0 + B(I)*(Y(I)+ Y(I+1))/2.0
     1 + C(I)*(T(I) + T(I+1))/2.0
      DC= AC*X(1) + BC*Y(1) + CC*T(1)
      DEN=AC**2+BC**2+CC**2
      XC= (DC*AC+D(1)*(CC*B(2)-BC*C(2)) + D(2)*(BC*C(1)-CC*B(1)))/DEN
      YC= (DC*BC+D(1)*(AC*C(2)-CC*A(2)) +D(2)*(CC*A(1)-AC*C(1)))/DEN
      ZC=(DC*CC+D(1)*(BC*A(2)-AC*B(2)) + D(2)*(AC*B(1)-BC*A(1)))/DEN
      AOAVG = (AAVE(1) + AAVE(2) + AAVE(3))/3.0
```

```
      UAVG = (UAVE(1) + UAVE(2) + UAVE(3))/3.0
      VAVG = (VAVE(1) + VAVE(2) + VAVE(3))/3.0
      DIST =  SQRTF((XC-X(1))**2+(YC-Y(1))**2+(ZC-T(1))**2)
      TESTA = 1.0E-10*DIST
      X(4) = DIST*UAVG/AOAVG + XC
      Y(4) = DIST*VAVG/AOAVG + YC
      T(4) = DIST*UCAP/AOAVG + ZC
C  THE OUTER ITERATION LOOP STARTS HERE.
      DO 300 L=1,90
  29  DO 13 I=1,3
      A(I) = UAVE(I)/UCAP
      B(I) = VAVE(I)/UCAP
      C(I) = AAVE(I)/UCAP
      UABC(I)= A(I)**2+B(I)**2-C(I)**2
      D(I)= X(I)-A(I)*T(I)
      E(I)=Y(I)-B(I)*T(I)
      F(I)=UABC(I)*T(I)-B(I)*Y(I)-A(I)*X(I)
  13  G(I)=X(I)**2 + Y(I)**2 + UABC(I)*(T(I)**2)
     1 -2.0*(X(I)*T(I)*A(I) + Y(I)*T(I)*B(I))
C  THE MIDDLE ITERATION LOOP STARTS HERE.  THE MACH CONE EQUATIONS ARE SOLVED IA
C  THE FOLLOWING STEPS.
      DO 200 M=1,90
      K = 0
  400 DO 201 I=1,3
      FX(I) = 2.0*(X(4)-A(I)*T(4) - D(I))
      FY(I) = 2.0*(Y(4) - B(I)*T(4) - E(I))
  201 FZ(I) = 2.0*(UABC(I)*T(4) - B(I)*Y(4) - A(I)*X(4) - F(I))
   62 DELTAX=0.0
      DELTAY=0.0
      DELTAZ=0.0
C  THE INNER ITERATION LOOP STARTS HERE.
   27 DO 14 I=1,3
       BCOL(I   )=-(X(4)**2+Y(4)**2+UABC(I)*(T(4)**2)-2.0*(A(I)*X(4)*T(4)
     1 + B(I)*Y(4)*T(4)   +       X(4)*D(I) + Y(4)*E(I) + T(4)*F(I))+G(I))
      AARRAY(I,1)=    FX(I)+    DELTAX-    A(I)*DELTAZ
      AARRAY(I,2)=    FY(I)+    DELTAY-    B(I)*DELTAZ
   14 AARRAY(I,3)=    FZ(I)+    UABC(I)*DELTAZ-    A(I)*DELTAX
     1 -    B(I)*DELTAY
      K = K + 1
      IF (K - 90) 60, 60, 61
   61 PRINT 190, X(4), Y(4), T(4)
  190 FORMAT (44H EXCESS LOOPING INNER LOOP IN FLOPT. X(4) = , E15.8,
     1 8H Y(4) = , E15.8, 8H T(4) = , E15.8)
      RETURN
   60 DET = 1.0
      N=XSIMEQF (3,3,1,AARRAY,BCOL,DET, M)
      GO TO (15, 16, 17), N
   16 PRINT 7
    7 FORMAT (24H OVER/UNDERFLOW IN FLOPT)
      RETURN
   17 PRINT 8
    8  FORMAT (28H AARRAY IS SINGULAR IN FLOPT)
      RETURN
   15 DO 18 I=1,3
      IF (AARRAY(I,1)) 18, 19 , 18
   18 CONTINUE
C  INNER LOOP CONVERGENCE TESTS FOLLOW.
      IF(ABSF(AARRAY(1,1))-TESTA) 20, 20, 21
   21 IF(ABSF(DELTAX-AARRAY(1,1))-DXTEST*ABSF(AARRAY(1,1))) 20, 20, 22
   20 IF(ABSF(AARRAY(2,1))-TESTA) 23, 23, 24
   24 IF(ABSF(DELTAY-AARRAY(2,1))-DXTEST*ABSF(AARRAY(2,1))) 23, 23, 22
   23 IF(ABSF(AARRAY(3,1))-TESTA) 19, 19, 26
   26 IF(ABSF(DELTAZ-AARRAY(3,1))-DXTEST*ABSF(AARRAY(3,1))) 19, 19, 22
   22 DELTAX=AARRAY(1,1)
      DELTAY= AARRAY(2,1)
      DELTAZ= AARRAY(3,1)
      IF(K-10)27,19,27
   19 X(4) = X(4) + AARRAY(1,1)
      Y(4) = Y(4) + AARRAY(2,1)
      T(4) = T(4) + AARRAY(3,1)
      IF(K-10)401,400,401
  401 DO 998 I=1,3
  998 RESIDU(I) = 1.0 - (2.0*(A(I)*X(4)*T(4) + B(I)*Y(4)*T(4) + D(I)*
     1 X(4) + E(I)*Y(4) + F(I)*T(4)) - G(I))/(X(4)**2 + Y(4)**2 + UABC(I
     2)*T(4)**2)
C  MIDDLE LOOP CONVERGENCE TESTS FOLLOW.
      DO 999 I=1,3
      IF (ABSF(RESIDU (I)) - XTEST) 999, 999, 200
  999 CONTINUE
      GO TO 91
  200 CONTINUE
      PRINT 191, X(4), Y(4), T(4)
  191 FORMAT (45H EXCESS LOOPING MIDDLE LOOP IN FLOPT. X(4) = , E15.8,
     1 8H Y(4) = , E15.8, 8H T(4) = , E15.8)
      RETURN
C  THE COORDINATE TRANSFORMATION IS CALCULATED NEXT.
   91 DO 92 I=1,3
      M(I) = UCAP/AAVE(I)
      E(I) = M(I)*(X(4) - X(I))/(T(4) - T(I))
      F(I) = M(I)*(Y(4) - Y(I))/(T(4) - T(I))
      A(I) = E(I) - UAVE(I)/AAVE(I)
      B(I) = F(I) - VAVE(I)/AAVE(I)
      D(I) = (-A(I)*UAVE(I) - B(I)*VAVE(I) - AAVE(I))/UCAP
      DEL(I) = A(I)*F(I) - B(I)*E(I)
      EL(I) = (B(I)*M(I) - D(I)*F(I))/DEL(I)
      EM(I) = (D(I)*E(I) - M(I)*A(I))/DEL(I)
      EK1(I) = 1.0/SQRTF(A(I)**2 + B(I)**2 + D(I)**2)
      EK2(I) = 1.0/SQRTF(E(I)**2 + F(I)**2 + M(I)**2)
      EK3(I) = 1.0/SQRTF(EL(I)**2 + EM(I)**2 + 1.0)
      BETA(I,1,1) = EK1(I)*A(I)
      BETA(I,1,2) = EK1(I)*B(I)
      BETA(I,1,3) = EK1(I)*D(I)
      BETA(I,2,1) = EK2(I)*E(I)
      BETA(I,2,2) = EK2(I)*F(I)
      BETA(I,2,3) = EK2(I)*M(I)
      BETA(I,3,1) = EK3(I)*EL(I)
      BETA(I,3,2) = EK3(I)*EM(I)
      BETA(I,3,3) = EK3(I)
C  THE NORMAL DERIVATIVES ARE NOW CALCULATED.
      AARRAY(I,1) = X(4) - X(I)
      AARRAY(I,2) = Y(4) - Y(I)
      AARRAY(I,3) = T(4) - T(I)
      BARRAY(I,1) = U(4) - U(I)
      BARRAY(I,2) = V(4) - V(I)
   92 BARRAY(I,3) = PK4 - P(I)
      DET = 1.0
      M = XSIMEQF(3, 3, 3, AARRAY, BARRAY, DET, M)
      GO TO (33, 34, 35), M
   34 PRINT 100
  100 FORMAT  (37H OVERFLOW NORMAL DERIVATIVES IN FLOPT)
      RETURN
   35 PRINT 101
```

```
 101  FORMAT (44H MATRIX SINGULAR NORMAL DERIVATIVES IN FLDPT)
      RETURN
C  THE COMPATIBILITY EQUATIONS ARE SOLVED NEXT.
 33   DO 36 I=1,3
      DO 36 J=1,3
      DDBETA(I,J) = AARRAY(1,J)*BETA(I,3,1) + AARRAY(2,J)*BETA(I,3,2)
     1    + AARRAY(3,J)*BETA(I,3,3)
 36   VDELBE(I,J) = UAVE(I)*BETA(I,J,1) + VAVE(I)*BETA(I,J,2)
     1    + UCAP*BETA(I,J,3)
      DO 37 I=1,3
      DO 38 J=1,2
 38   BARRAY(I,J) = BETA(I,2,J)*VDELBE(I,1) - BETA(I,1,J)*VDELBE(I,2)
      BARRAY(I,3) = (VDELBE(I,1)*VDELBE(I,2)/AAVE(I)**2 - BETA(I,2,1)*BE
     1  TA(I,1,1) - BETA(I,2,2)*BETA(I,1,2))/ROEAVE(I)
      E(I) = SQRTF((X(6)-X(I))**2+(Y(6)-Y(I))**2+(T(6)-T(I))**2)
      AARRAY(I,1) = U(I)*BARRAY(I,1)+V(I)*BARRAY(I,2)+P(I)*BARRAY(I,3)
      F(I) = VDELBE(I,1)*(DDBETA(I,1)*BETA(I,3,1) +DDBETA(I,2)*BETA(I,3,
     12))-VDELBE(I,3)*(DDBETA(I,1)*BETA(I,1,1)+DDBETA(I,2)*BETA(I,1,2))
     2 +DDBETA(I,3)*(VDELBE(I,3)*VDELBE(I,1)/AAVE(I)**2-BETA(I,3,1)*BETA
     3 (I,1,1)-BETA(I,3,2)*BETA(I,1,2))/ROEAVE(I)
 37   AARRAY(I,1) = AARRAY(I,1) - E(I)*F(I)
      DET = 1.0
      M = XSIMEQF (3, 3, 1, BARRAY, AARRAY, DET, M)
      GO TO (39, 40, 41), M
 40   PRINT 102
 102  FORMAT (42H OVERFLOW COMPATIBILITY EQUATIONS IN FLDPT)
      RETURN
 41   PRINT 103
 103  FORMAT (49H MATRIX SINGULAR COMPATIBILITY EQUATIONS IN FLDPT)
      RETURN
 39   U(6) = BARRAY(1,1)
      V(6) = BARRAY(2,1)
      P(6) = BARRAY(3,1)
C  THE STREAMLINE IS PROJECTED BACK TO DETERMINE THE ENTROPY NEXT.
 43   UAVE5 = (U(6) + U(5))/2.0
      VAVE5 = (V(6) + V(5))/2.0
      GAMMA(1) = Y(2)*T(3)+Y(1)*T(2)+Y(3)*T(1)-Y(2)*T(1)-Y(3)*T(2)-Y(1)*
     1T(3)
      GAMMA(2) = X(1)*T(3)+X(2)*T(1)+X(3)*T(2)-X(3)*T(1)-X(1)*T(2)-X(2)*
     1T(3)
      GAMMA(3) = X(1)*Y(2)+X(2)*Y(3)+X(3)*Y(1)-X(3)*Y(2)-X(1)*Y(3)-X(2)*
     1Y(1)
      GAMMA( 4) = X(1)*Y(2)*T(3)+X(2)*Y(3)*T(1)+X(3)*Y(1)*T(2)
     1-X(3)*Y(2)*T(1)-X(1)*Y(3)*T(2)-X(2)*Y(1)*T(3)
      BAG = UAVE5*GAMMA(1) + VAVE5*GAMMA(2) + UCAP*GAMMA(3)
      X(5) = ((X(6)*VAVE5-Y(6)*UAVE5)*GAMMA(2)+(X(6)*UCAP-T(6)*UAVE5)*
     1 GAMMA(3)+UAVE5*GAMMA(4))/BAG
      Y(5) = ((Y(6)*UCAP-T(6)*VAVE5)*GAMMA(3)+(Y(6)*UAVE5-X(6)*VAVE5)*
     1 GAMMA(1)+VAVE5*GAMMA(4))/BAG
      T(5) = ((T(6)*UAVE5-X(6)*UCAP)*GAMMA(1)+(T(6)*VAVE5-Y(6)*UCAP)*GAM
     1MA(2)+UCAP*GAMMA(4))/BAG
      CALL XINTER (X, Y, T, U, V, P, S)
      S(6) = S(5)
      ROE6 = ROEDET (P(6), S(6))
      A6 = ADET (P(6), S(6))
C  OUTER LOOP CONVERGENCE TESTS FOLLOW.
      VTOT = SQRTF(U(6)**2 + V(6)**2)
      IF (P(6) - VTOT) 800, 801, 801
 801  IF (ABSF(P(6) - PK6) - VTEST*P(6)) 48, 48, 45
 800  IF (ABSF(VTOT - VTOTK) - VTEST*VTOT) 48, 48, 45
C  AVERAGE VALUES ARE CALCULATED BEFORE REPEATING THE ITERATION.
 45   DO 49 I=1,3
      UAVE(I) = (U(I) + U(6))/2.0
      VAVE(I) = (V(I) + V(6))/2.0
      PAVE(I) = (P(I) + P(6))/2.0
      SAVE(I) = (S(I) + S(6))/2.0
      ROEAVE(I) = ROEDET (PAVE(I), SAVE(I))
 49   AAVE(I) = ADET (PAVE(I), SAVE(I))
      PK6 = P(6)
 300  VTOTK = VTOT
 65   PRINT 132, X(6), Y(6), T(6)
 132  FORMAT (44H EXCESS LOOPING OUTER LOOP IN FLDPT, X(6) = , E15.8, 8H
     1 Y(6) = , E15.8, 8H T(6) = , E15.8)
 48   RETURN
      END



      SUBROUTINE SMKPT
C  THIS SUBROUTINE CALLS THE FUNCTIONS ADET AND ROEDET.
      DIMENSION X(5),Y(5), T(5), U(5), V(5), P(5), S(5), ROE(3), ASPD(3),
     2UAVE(3), VAVE(3), AAVE(3), XB(3), YB(3), TB(3), PAVE(3), SAVE(3),
     3ROEAVE(3), AARRAY(3,3), M(3), A(2), B(2), C(2), UABC(2), D(2),
     4E(2), F(2), G(2), FX(2), FY(2), FZ(2), DEL(2), EL(2), EM(2),
     5EK1(2), EK2(2), EK3(2), BETA(2,3,3), BARRAY(3,3), DDBETA(2,3),
     6VDELBE(2,3), CARRAY(2,2), BCCEF(4), RESIDU(3), MS(3), FLPTS1(50,
     720, 7), BDPTS1(50, 7), SKPTS, (50, 10)
      COMMON X, Y, T, U, V, A6, UCAP, P, S, ROE6, DXTEST, XTEST, VTEST,
     1BCOEF, ENX6, ENY6, ENT6, ENX5, ENY5, ENT5, FLPTS1,BDPTS1,SKPTS1,
     2TEST,UPLUS,VPLUS,PPLUS,SPLUS
      EQUIVALENCE (C(1),CARRAY(1)),(UABC(1),CARRAY(3)),(DETA,OCAP),(U3,P
     1M1),(V3,ACAP),(AO3,BCAP),(FZ(1),DEL(1)),(M(1),EL(1)),
     2(A1,EMPLSQ),(C1,CCAP),(D1,DCAP),(ABC,QNP),(VCAP,QNM),(VDOTN,ECAP)
     3,(TERM,FCAP),(DELTAX,GCAP),(DELTAY,UNX),(DELTAZ,UNY),(FX(1),VNX),
     4(FX(2),VNY),(FY(1),PNX),(FY(2),PNY),(YB(1),EM(1)),(YB(3),EK1(1)),
     5(TB(1), EK1(2)), (TB(2), EK2(1)), (VTOT,DDBETA(1)),(DET,DDBETA(2))
C  PRELIMINARY CALCULATIONS OF AVERAGES START HERE.
      DO 10 I=1,3
      ROE(I) = ROEDET (P(I), S(I))
      ASPD(I) = ADET (P(I), S(I))
      UAVE(I) = U(I)
      VAVE(I) = V(I)
      ROEAVE(I) = ROE(I)
 10   AAVE(I) = ASPD(I)
      P6 = P(3)
      VTOTK = SQRTF(U(3)**2 + V(3)**2)
      APLUS = ADET(PPLUS, SPLUS)
      ENX5 = ENX6
      ENY5 = ENY6
      ENT5 = ENT6
      ENX65 = (ENX6 + ENX5)/2.0
      ENY65 = (ENY6 + ENY5)/2.0
      ENT65 = (ENT6 + ENT5)/2.0
C  CALCULATION OF THE FIRST ESTIMATE OF THE COORDINATES FOLLOWS.
      B6 = ENX65*X(3) + ENY65*Y(3) + ENT65*T(3)
      DETA = ENX65**2 + ENY65**2 + ENT65**2
      DO 9 I=1,2
      XB(I) = (ENX65*B6 + (ENY65**2 + ENT65**2)*X(I) - ENX65*ENY65*Y(I)
     1  - ENX65*ENT65*T(I))/DETA
      YB(I) = (ENY65*B6 + (ENT65**2 + ENX65**2)*Y(I) - ENY65*ENT65*T(I)
     1 - ENY65*ENX65*X(I))/DETA
```

```
 9    YB(I) = (ENT6S*B4 + (ENX6S**2 + ENY6S**2)*T(I) - ENT6S*ENX6S*X(I)
     1 - ENT6S*ENY6S*Y(I))/DETA
      XB(3) = (XB(1) + XB(2))/2.0
      YB(3) = (YB(1) + YB(2))/2.0
      TB(3) = (TB(1) + TB(2))/2.0
      U3 = (UAVE(1) + UAVE(2))/2.0
      V3 = (VAVE(1) + VAVE(2))/2.0
      AO3 = (AAVE(1) + AAVE(2))/2.0
      A1  = U3 - (U3*ENX6S**2 + V3*ENX6S*ENY6S + UCAP*ENX6S*ENT6S)/DETA
      C1 = V3 - (U3*ENX6S*ENY6S + V3*ENY6S**2 + UCAP*ENY6S*ENT6S)/DETA
      D1 = UCAP - (U3*ENX6S*ENT6S + V3*ENY6S*ENT6S + UCAP*ENT6S**2)/DETA
      ABC=SQRTF(A1**2+C1**2+D1**2)
      DIST=SQRTF((XB(3)-(X(1)+X(2))/2.0)**2+(YB(3)-(Y(1)+Y(2))/2.0)**2
     1+(TB(3)-(T(1)+T(2))/2.0)**2)
      TESTA = 1.0E-10*DIST
      ETEST = XTEST*10.0
      VCAP=SQRTF(U 3 **2+V 3 **2+UCAP**2)
      VDOTN = (U3*ENX6S + V3*ENY6S + UCAP*ENT6S)/VCAP
      TERM=(1.0+(AO 3 /UCAP)*TANF(ASINF(VDOTN)))/(AO 3 /UCAP-
     1TANF(ASINF(VDOTN)))
      X(6) = XB(3) + TERM*DIST*A1/ABC
      Y(6) = YB(3) + TERM*DIST*C1/ABC
      T(6) = TB(3) + TERM*DIST*D1/ABC
C  THE OUTER ITERATION LOOP STARTS HERE.
      DO 900 L=1,50
      DO 11 I=1,2
      A(I) = UAVE(I)/UCAP
      B(I) = VAVE(I)/UCAP
      C(I) = AAVE(I)/UCAP
      UABC(I)= A(I)**2+B(I)**2-C(I)**2
      D(I)= X(I)-A(I)*T(I)
      E(I)=Y(I)-B(I)*T(I)
      F(I)=UABC(I)*T(I)-B(I)*Y(I)-A(I)*X(I)
 11   G(I)=X(I)**2 + Y(I)**2 + UABC(I)*(T(I)**2)
     1 -2.0*(X(I)*T(I)*A(I) + Y(I)*T(I)*B(I))
C  THE MIDDLE ITERATION LOOP STARTS HERE.  THE MACH CONE AND SHOCK WAVE SURFACE
C  EQUATIONS ARE SOLVED IN THE FOLLOWING STEPS.
      DO 201 N=1,50
       DO 200 I=1,2
      MS(I )=-(X(6)**2+Y(6)**2+UABC(I)*(T(6)**2)-2.0*(A(I)*X(6)*T(6)+B(I
     1)*Y(6)*T(6) +      X(6)*D(I)+Y(6)*E(I)+T(6)*F(I))+G(I))
      FX(I) = 2.0*(X(6)-A(I)*T(6)-D(I))
      FY(I) = 2.0*(Y(6)-B(I)*T(6)-E(I))
 200  FZ(I) = 2.0*(UABC(I)*T(6)-B(I)*Y(6)-A(I)*X(6)-F(I))
 49   DELTAX=0.0
      DELTAY=0.0
      DELTAZ=0.0
      MS(3 )=(ENX6S*(X(3)-X(6))+ENY6S*(Y(3)-Y(6))+ENT6S*(T(3)-T(6)))
C  THE INNER ITERATION LOOP STARTS HERE.
      DO 29 K=1,50
      DO 12 I=1,2
      M(I)=MS(I)
      AARRAY(1,1)=    FX(I)*    DELTAX-    A(I)*DELTAZ
      AARRAY(1,2)=    FY(I)*    DELTAY-    B(I)*DELTAZ
   12 AARRAY(1,3)=    FZ(I)*    UABC(I)*DELTAZ-    A(I)*DELTAX
     1 -    B(I)*DELTAY
      M(3)=MS(3)
      AARRAY(3,1)=ENX6S
      AARRAY(3,2)=ENY6S
      AARRAY(3,3)=ENT6S
 91   DET = 1.0
      N = XSIMEQF (3, 3, 1, AARRAY,M,DET,XO)
      GO TO (13, 14, 15), N
 14   PRINT 7
    7 FORMAT (24H OVER/UNDERFLOW IN SHKPT)
      RETURN
 15   PRINT 8
    8  FORMAT  28H AARRAY IS SINGULAR IN SHKPT)
      RETURN
C  INNER LOOP CONVERGENCE TESTS FOLLOW.
 13   IF(ABSF(AARRAY(1,1))-TESTA) 16, 16, 17
 17   IF(ABSF(DELTAX-AARRAY(1,1))-DXTEST*ABSF(AARRAY(1,1))) 16, 16, 18
 16   IF(ABSF(AARRAY(2,1))-TESTA) 19, 19, 20
 20   IF(ABSF(DELTAY-AARRAY(2,1))-DXTEST*ABSF(AARRAY(2,1))) 19, 19, 18
 19   IF(ABSF(AARRAY(3,1))-TESTA) 21, 21, 22
 22   IF(ABSF(DELTAZ-AARRAY(3,1))-DXTEST*ABSF(AARRAY(3,1))) 21, 21, 18
 18   DELTAX= AARRAY(1,1)
      DELTAY= AARRAY(2,1)
 23   DELTAZ= AARRAY(3,1)
 52   PRINT 107, X(6), Y(6), T(6)
 107  FORMAT (44H EXCESS LOOPING INNER LOOP IN SHKPT, X(6) = , E15.8,
     1 8H Y(6) = , E15.8, 8H T(6) = , E15.8)
      RETURN
 21   X(6) = X(6) + AARRAY(1,1)
      Y(6) = Y(6) + AARRAY(2,1)
      T(6) = T(6) + AARRAY(3,1)
C  RESIDUALS ARE CALCULATED NEXT.
      DO 998 I=1,2
 998  RESIDU(I) = 1.0 - (2.0*(A(I)*X(6)*T(6) + B(I)*Y(6)*T(6) + D(I)*X(6
     1) + E(I)*Y(6) + F(I)*T(6)) - G(I))/(X(6)**2 + Y(6)**2 + UABC(I)*T(
     26)**2)
 997  RESIDU(3)=MS(3    ) / (ABSF(X(3) - X(6)) + ABSF(Y(3) - Y(6)) + ABSF
     1(T(3) - T(6)))
C  MIDDLE LOOP CONVERGENCE TESTS FOLLOW.
      DO 995 I=1,3
      IF (ABSF(RESIDU(I)) - XTEST) 995, 995, 201
 995  CONTINUE
      GO TO 27
 201  CONTINUE
 50   PRINT 106, X(6) , Y(6), T(6)
 106  FORMAT (49H EXCESS LOOPING MIDDLE LOOP IN SHKPT, X(6) = , E15.8,
     1 8H Y(6) = , E15.8, 8H T(6) = , E15.8)
      RETURN
C  THE SHOCK WAVE EQUATIONS ARE SOLVED NEXT.
 27   PHI = 1.0/SQRTF(1.0 - ENT5**2)
      QNP = (UPLUS*ENX5 + VPLUS*ENY5 + UCAP*ENT5)*PHI
      EMPLSQ = (QNP/APLUS)**2
      QNM = QNP*(0.4*EMPLSQ + 2.0)/(2.4*EMPLSQ)
      P(6)   = PPLUS*(1.0 + 1.1666667*(EMPLSQ-1.0))
      S(6)   = SPLUS + 0.068806*LOGF((P(6)  /PPLUS)**2.5*(QNM/QNP)**3.5)
      U(6) = UPLUS+ ENX5*PHI*(QNM-QNP)
      V(6) = VPLUS+ ENY5*PHI*(QNM-QNP)
      ROE6   = ROEDET (P(6), S(6))
      A6     = ADET (P(6) , S(6))
C  THE COORDINATE TRANSFORMATION IS CALCULATED NEXT.
      DO 28 I=1,2
      G(I) = UCAP/AAVE(I)
      E(I) = G(I)*(X(6) - X(I))/(T(6) - T(I))
      F(I) = G(I)*(Y(6) - Y(I))/(T(6) - T(I))
      A(I) = E(I) - UAVE(I)/AAVE(I)
      B(I) = F(I) - VAVE(I)/AAVE(I)
      D(I) = (-A(I)*UAVE(I) - B(I)*VAVE(I) - AAVE(I))/UCAP
```

```
      DEL(I) = A(I)*F(I) - B(I)*E(I)
      EL(I) = (B(I)*G(I) - D(I)*F(I))/DEL(I)
      EM(I) = (D(I)*E(I) - G(I)*A(I))/DEL(I)
      EK1(I) = 1.0/SQRTF(A(I)**2 + B(I)**2 + D(I)**2)
      EK2(I) = 1.0/SQRTF(E(I)**2 + F(I)**2 + G(I)**2)
      EK3(I) = 1.0/SQRTF(EL(I)**2 + EM(I)**2 + 1.0)
      BETA(I,1,1) = EK1(I)*A(I)
      BETA(I,1,2) = EK1(I)*B(I)
      BETA(I,1,3) = EK1(I)*D(I)
      BETA(I,2,1) = EK2(I)*E(I)
      BETA(I,2,2) = EK2(I)*F(I)
      BETA(I,2,3) = EK2(I)*G(I)
      BETA(I,3,1) = EK3(I)*EL(I)
      BETA(I,3,2) = EK3(I)*EM(I)
   20 BETA(I,3,3) = EK3(I)
C  THE NORMAL DERIVATIVES ARE NOW CALCULATED.
      DO 29 I=1,3
      AARRAY(I,1) = X(4) - X(I)
      AARRAY(I,2) = Y(4) - Y(I)
      AARRAY(I,3) = T(4) - T(I)
      BARRAY(I,1) = U(4) - U(I)
      BARRAY(I,2) = V(4) - V(I)
   29 BARRAY(I,3) = P(4) - P(I)
      DET = 1.0
      M = RSIMEQF(3, 3, 3, AARRAY, BARRAY, DET, RB)
      GO TO (30, 31, 32), M
   31 PRINT 100
  100 FORMAT  (37H OVERFLOW NORMAL DERIVATIVES IN SHKPT)
      RETURN
   32 PRINT 101
  101 FORMAT (44H MATRIX SINGULAR NORMAL DERIVATIVES IN SHKPT)
      RETURN
C  THE COMPATIBILITY EQUATIONS ARE NOW SOLVED FOR THE NORMAL VECTOR COMPONENT
C  CORRECTIONS.
   30 DO 33 I=1,2
      DO 33 J=1,3
      DDBETA(I,J) = AARRAY(1,J)*BETA(I,3,1) + AARRAY(2,J)*BETA(I,3,2)
     1    + AARRAY(3,J)*BETA(I,3,3)
   33 VDELBE(I,J) = UAVE(I)*BETA(I,J,1) + VAVE(I)*BETA(I,J,2)
     1    + UCAP*BETA(I,J,3)
      IF (ABSF(ENX5) - ABSF(ENY5)) 900, 900, 901
  900 ENA = ENX5
      ENB = ENY5
      UVA = UPLUS
      UVB = VPLUS
      IDETRM = 1
      GO TO 902
  901 ENA = ENY5
      ENB = ENX5
      UVA = VPLUS
      UVB = UPLUS
      IDETRM = 2
  902 ACAP = PHI*(QNM-QNP)
      BCAP = 0.83333333*PHI**2*(EMPLSQ + 1.0)*(UVB*ENA/ENB - UVA)/EMPLSQ
      ECAP = PHI*(UCAP - UVB*ENT5/ENB + ENT5*PHI*QNP)
      CCAP = PHI**2*(                    ENT5*PHI*(QNM - QNP) - 0.83333
     133333*(EMPLSQ + 1.0)*(ECAP/PHI)/EMPLSQ)
      DCAP = PHI*(UVA-UVB*ENA/ENB)
      GCAP = 2.9333333*PPLUS*SQRTF(EMPLSQ)/APLUS
      UANA = ACAP + ENA*BCAP
      UANT = ENA*CCAP
      UBNA = ENB*BCAP - ACAP*ENA/ENB
      UBNT = ENB*CCAP - ACAP*ENT5/ENB
      PNX = DCAP*GCAP
      PNY = ECAP*GCAP
      GO TO (903, 904), IDETRM
  903 UNX = UANA
      UNY = UANT
      VNX = UBNA
      VNY = UBNT
      GO TO 905
  904 VNX = UANA
      VNY = UANT
      UNX = UBNA
      UNY = UBNT
  905 DO 34 I=1,2
      DO 35 J=1,2
   35 BARRAY(I,J) = BETA(I,2,J)*VDELBE(I,1) - BETA(I,1,J)*VDELBE(I,2)
      BARRAY(I,3) = (VDELBE(I,1)*VDELBE(I,2)/AAVE(I)**2 - BETA(I,2,1)*BE
     1  TA(I,1,1) - BETA(I,2,2)*BETA(I,1,2))/ROEAVE(I)
      CARRAY(I,1) = UNX*BARRAY(I,1)+VNX*BARRAY(I,2)+PNX*BARRAY(I,3)
      CARRAY(I,2) = UNY*BARRAY(I,1)+VNY*BARRAY(I,2)+PNY*BARRAY(I,3)
      AARRAY(I,1) = (U(I)-U(4))*BARRAY(I,1) + (V(I)-V(4))*BARRAY(I,2)
     1 + (P(I)-P(4))*BARRAY(I,3)
      E(I) = SQRTF((X(4)-X(I))**2+(Y(4)-Y(I))**2+(T(4)-T(I))**2)
      F(I) = VDELBE(I,1)*(DDBETA(I,1)*BETA(I,3,1) +DDBETA(I,2)*BETA(I,3,
     12))-VDELBE(I,3)*(DDBETA(I,1)*BETA(I,1,1)+DDBETA(I,2)*BETA(I,2,2))
     2 +DDBETA(I,3)*(VDELBE(I,3)*VDELBE(I,1)/AAVE(I)**2-BETA(I,3,1)*BETA
     3 (I,1,1)-BETA(I,3,2)*BETA(I,2,2))/ROEAVE(I)
   34 AARRAY(I,1) = AARRAY(I,1) - E(I)*F(I)
      DET = 1.0
      M = RSIMEQF(2, 2, 1, CARRAY, AARRAY, DET, M)
      GO TO (36, 37, 38), M
   37 PRINT 102
  102 FORMAT (42H OVERFLOW COMPATIBILITY EQUATIONS IN SHKPT)
      RETURN
   38 PRINT 103
  103 FORMAT (49H MATRIX SINGULAR COMPATIBILITY EQUATIONS IN SHKPT)
      RETURN
   36 GO TO (800, 801), IDETRM
  800 IF (ENY5) 802, 802, 803
  802 DELNY=-SQRTF(1.0-(ENX5*CARRAY(1,1))**2-(ENT5*CARRAY(2,1))**2)-ENY5
      GO TO 807
  803 DELNY= SQRTF(1.0-(ENX5*CARRAY(1,1))**2-(ENT5*CARRAY(2,1))**2)-ENY5
  807 DELNX = CARRAY(1,1)
      GO TO 805
  801 IF (ENX5) 804, 804, 806
  804 DELNX=-SQRTF(1.0-(ENY5*CARRAY(1,1))**2-(ENT5*CARRAY(2,1))**2)-ENX5
      GO TO 808
  806 DELNX= SQRTF(1.0-(ENY5*CARRAY(1,1))**2-(ENT5*CARRAY(2,1))**2)-ENX5
  808 DELNY = CARRAY(1,1)
  805 ENX5 = ENX5 + DELNX
      ENY5 = ENY5 + DELNY
      ENT5 = ENT5 + CARRAY(2,1)
C  OUTER LOOP CONVERGENCE TESTS FOLLOW.
   39 IF (ABSF (DELNX)- 1.0E-7) 901, 901, 902
  902 IF (ABSF (DELNX       ) - ETEST*ABSF (ENX5)) 901, 901, 43
  901 IF (ABSF (DELNY)- 1.0E-7) 903, 903, 904
  904 IF (ABSF (DELNY       ) - ETEST*ABSF (ENY5)) 903, 903, 43
  903 IF (ABSF (CARRAY(2,1)) - 1.0E-7) 905, 905, 906
  906 IF (ABSF(CARRAY(2,1)) - ETEST*ABSF(ENT5)) 905, 905, 43
  905 VTOT = SQRTF(U(4)**2 + V(4)**2)
```

```
      IF (VTOT - P(6)) 41, 41, 42
 41   IF (ABSF(P(4) - P4) - VTEST*P(6)) 47, 47, 43
 42   IF (ABSF(VTOT - VTOTK) - VTEST*VTOT) 47, 47, 43
C  AVERAGE VALUES ARE CALCULATED BEFORE REPEATING THE ITERATION.
   43 DO 48 I=1,3
      UAVE(I) = (U(I) + U(6))/2.0
      VAVE(I) = (V(I) + V(6))/2.0
      PAVE(I) = (P(I) + P(6))/2.0
      SAVE(I) = (S(I) + S(6))/2.0
      ENN45 = (ENN4 + ENN5)/2.0
      ENV45 = (ENV4 + ENV5)/2.0
      ENT45 = (ENT4 + ENT5)/2.0
      ROEAVE(I) = ROEDET (PAVE(I), SAVE(I))
 48   AAVE(I) = ADET (PAVE(I), SAVE(I))
      P4 = P(6)
 300  VTOTK = VTOT
 49   PRINT 104, X(6) , Y(6), T(6)
 104  FORMAT (44H EXCESS LOOPING OUTER LOOP IN SMKPT. X(6) = , E15.8,
     1 8H Y(6) = , E15.8, 8H T(6) = , E15.8)
 47   RETURN
      END



      SUBROUTINE BDYPT
C  THIS SUBROUTINE CALLS RCHANG AND THE FUNCTIONS ADET AND ROEDET.
      DIMENSION X(9), Y(9), T(9), U(9), V(9), P(9), S(9), BCOE(6), XA(3)
     1, YA(3), TA(3), UAVE(2), VAVE(2), PAVE(2), SAVE(2), AAVE(2),
     2ROEAVE(2), AARRAY(3,3), BCOL(3  ), ARGB(3), A(2), B(2), C(2), UABC
     3(2), D(2), E(2), F(2), G(2), H(2), FX(2), FY(2), FZ(2), BETA(3,3,3
     4), BARRAY(3,3), DDBETA(3,3), VDELBE(3,3), RESIDU(3)
      COMMON X,Y,T,U,V,A4,UCAP,P,S,ROE4,DXTEST,XTEST,VTEST,BCOE ,ENN4,
     1ENV4,ENT4,ENN5,ENV5,ENT5
      EQUIVALENCE (DETA, BETA(1), U4DUM), (XA(1), BETA(2), V4DUM),
     1(XA(2),AA2), (XA(3), BB2), (YA(1), BETA(5)), (TA(1), BETA(8)),
     2(UAV, BETA(11)), (VAV, BETA(12)), (AAV, BETA(13)), (ARAG, BETA
     3(26)), (A1, BETA(14)), (C1, BETA(15)), (D1, BETA(16)), (ABC, BETA(
     417)), (DIST, BETA(18)), (VCAP, BETA(19)), (DELTAX, BETA(20)), (DEL
     5TAY, BETA(21)), (DELTAZ, BETA(22)), (BCOL(1), BETA(23)),
     6(ARGB(1), BETA(27)), (ARGB(2), DDBETA(1)), (VTOT, DDBETA(3))
C  PRELIMINARY CALCULATIONS OF AVERAGES START HERE.
      U4 = (U(1) + U(2))/2.0
      V4 = (V(1) + V(2))/2.0
      U(5) = (U(3) + U(4))/2.0
      V(5) = (V(3) + V(4))/2.0
      P4PRE = (P(3) + P(4))/2.0
      VTOTK = SQRTF ((U(3) + U(4))**2 + (V(3) + V(4))**2)/2.0
      DO 1 I=1,2
       UAVE(I) = U(I)
       VAVE(I) = V(I)
       PAVE(I) = P(I)
       SAVE(I) = S(I)
       ROEAVE(I) = ROEDET(P(I), S(I))
 1     AAVE(I) = ADET(P(I), S(I))
C  CALCULATION OF THE FIRST ESTIMATE OF THE COORDINATES FOLLOWS.
      CALL RCHANG(X,Y,T,UCAP,AARRAY,BCOL,B1,B2,B3,B4,DELTAX,DELTAY,DELTA
     12, 1, N, RESIDU)
  130 DETA=B1**2+B2**2+B3**2
      DO 101 I=1,2
      XA(I) =(B1*B4+(B2**2+B3**2)*X(I)-B1*B2*Y(I)-B1*B3*T(I))/DETA
      YA(I) =(B2*B4+(B3**2+B1**2)*Y(I)-B2*B3*T(I)-B2*B1*X(I))/DETA
 101  TA(I) =(B3*B4+(B1**2+B2**2)*T(I)-B3*B1*X(I)-B3*B2*Y(I))/DETA
       XA(3) = (XA(1) + XA(2))/2.0
       YA(3) = (YA(1) + YA(2))/2.0
 125   TA(3) = (TA(1) + TA(2))/2.0
      UAV = (UAVE(1) + UAVE(2))/2.0
       VAV = (VAVE(1) + VAVE(2))/2.0
      AAV = (AAVE(1) + AAVE(2))/2.0
      A1=UAV -(UAV *B1**2+VAV *B1*B2+UCAP*B1*B3)/DETA
      C1=VAV -(UAV *B1*B2+VAV *B2**2+UCAP*B2*B3)/DETA
      D1=UCAP-(UAV *B1*B3+VAV *B2*B3+UCAP*B3**2)/DETA
      ABC=SQRTF(A1**2+C1**2+D1**2)
      DIST = SQRTF((XA(3) - (X(1) + X(2))/2.0)**2 + (YA(3) - (Y(1) +
     1 Y(2))/2.0)**2 + (TA(3) - (T(1) + T(2))/2.0)**2)
      VCAP=SQRTF(UAV **2+VAV **2+UCAP**2)
      VELDN = (UAV*B1 + VAV*B2 + UCAP*B3)/(VCAP*SQRTF(DETA))
      IF(VELDN)305,306,306
 305  VELDN=0.0
 306  X(5) = XA(3) + DIST*A1*(VCAP/AAV + VELDN)/ABC
      Y(5) = YA(3) + DIST*C1*(VCAP/AAV + VELDN)/ABC
 131  T(5) = TA(3) + DIST*D1*(VCAP/AAV + VELDN)/ABC
      TESTA = 1.0E-10*DIST
C  THE OUTER ITERATION LOOP STARTS HERE.
      DO 11 L=1,50
      DO 102 I=1,2
       A(I) = UAVE(I)/UCAP
       B(I) = VAVE(I)/UCAP
       C(I) = AAVE(I)/UCAP
      UABC(I)= A(I)**2+B(I)**2-C(I)**2
      D(I)= X(I)-A(I)*T(I)
      E(I)=Y(I)-B(I)*T(I)
      F(I)=UABC(I)*T(I)-B(I)*Y(I)-A(I)*X(I)
 102  G(I)=X(I)**2 + Y(I)**2 + UABC(I)*(T(I)**2)
     1 -2.0*(X(I)*T(I)*A(I) + Y(I)*T(I)*B(I))
C  THE MIDDLE ITERATION LOOP STARTS HERE.  THE MACH CONE AND BODY SURFACE
C  EQUATIONS ARE SOLVED IN THE FOLLOWING STEPS.
      DO 140 K=1,50
      DO 150 I=1,2
      H(I) = X(5)**2 + Y(5)**2 + UABC(I)*(T(5)**2) - 2.0*(A(I)*X(5)*T(5)
     1 + B(I)*Y(5)*T(5))- 2.0*(X(5)*D(I) + Y(5)*E(I) + T(5)*F(I)) + G(I)
      FX(I) = 2.0*(X(5) - A(I)*T(5) - D(I))
      FY(I) = 2.0*(Y(5) - B(I)*T(5) - E(I))
 150  FZ(I) = 2.0*(UABC(I) * T(5) - B(I)*Y(5) - A(I)*X(5) - F(I))
      DELTAX=0.0
      DELTAY=0.0
      DELTAZ=0.0
C  THE INNER ITERATION LOOP STARTS HERE.
      DO 113 N=1,50
      DO 103 I=1,2
      AARRAY(I,1)=     FX(I)*     DELTAX-     A(I)*DELTAZ
      AARRAY(I,2)=     FY(I)*     DELTAY-     B(I)*DELTAZ
      AARRAY(I,3)=     FZ(I)*     UABC(I)*DELTAZ-     A(I)*DELTAX
     1 -     B(I)*DELTAY
 103  BCOL(I  )= -     H(I)
      CALL RCHANG(X,Y,T,UCAP,AARRAY,BCOL,B1,B2,B3,B4,DELTAX,DELTAY,DELTA
     12, 2, N, RESIDU)
      ARGA=1.0
 132  M=SIMEQF (3,3,1,AARRAY,BCOL,ARGA,ARGB)
      GO TO (106,104,105),M
 104  PRINT 107
 107  FORMAT (39H OVER/UNDERFLOW COORDINATE EQS IN BDYPT)
```

```
      RETURN
  109 PRINT 108
  108  FORMAT (40H MATRIX SINGULAR COORDINATE EQS IN BDYPT)
      RETURN
C  INNER LOOP CONVERGENCE TESTS FOLLOW.
  106  IF (ABSF(AARRAY(1,1)) - TESTA) 110, 110, 122
  122  IF (ABSF(DELTAX - AARRAY(1,1)) - DXTEST*ABSF(AARRAY(1,1))) 110,
     1 110, 109
  110  IF (ABSF(AARRAY(2,1)) - TESTA) 111, 111, 123
  123  IF (ABSF(DELTAY - AARRAY(2,1)) - DXTEST*ABSF(AARRAY(2,1))) 111,
     1 111, 109
  111  IF (ABSF(AARRAY(3,1)) - TESTA) 112, 112, 124
  124  IF (ABSF(DELTAZ - AARRAY(3,1)) - DXTEST*ABSF(AARRAY(3,1))) 112,
     1 112, 109
  109 DELTAX=AARRAY(1,1)
      DELTAY= AARRAY(2,1)
  113  DELTAZ= AARRAY(3,1)
  102 PRINT 103, X(5), Y(5), T(5)
  103   FORMAT (44H EXCESS LOOPING INNER LOOP IN BDYPT, X(4) = , E15.8,
     1 8H Y(4) = , E15.8, 8H T(4) = , E15.8)
      RETURN
  112  X(5) = X(5) + AARRAY(1,1)
      Y(5) = Y(5) + AARRAY(2,1)
  133  T(5) = T(5) + AARRAY(3,1)
C  RESIDUALS ARE CALCULATED NEXT.
      DO 999 I=1,2
  999  RESIDU(I) = M(I)/(X(5)**2 + Y(5)**2 + UABC(I)*T(5)**2)
      CALL XCHANG(X,Y,T,UCAP,AARRAY,BCOL,B1,B2,B3,B4,DELTAX,DELTAY,DELTA
     12, 3, N, RESIDU)
C  MIDDLE LOOP CONVERGENCE TESTS FOLLOW.
  144  DO 998 I=1,3
      IF (ABSF(RESIDU(I)) - XTEST) 998, 998, 140
  998  CONTINUE
      GO TO 115
  140  CONTINUE
      PRINT 146, X(5), Y(5), T(5)
  146   FORMAT (45H EXCESS LOOPING MIDDLE LOOP IN BDYPT, X(4) = , E15.8,
     1 8H Y(4) = , E15.8, 8H T(4) = , E15.8)
      RETURN
  115  CALL XCHANG(X,Y,T,UCAP,AARRAY,BCOL,B1,B2,B3,B4,DELTAX,DELTAY,DELTA
     12, 4, N, RESIDU)
C  THE COORDINATE TRANSFORMATION IS CALCULATED NEXT.
      DO 210 I=1,2
      M(I) = UCAP/AAVE(I)
      E(I) = M(I)*(X(5) - X(I))/(T(5) - T(I))
      F(I) = M(I)*(Y(5) - Y(I))/(T(5) - T(I))
      A(I) = E(I) - UAVE(I) / AAVE(I)
      B(I)  = F(I) - VAVE(I) / AAVE(I)
      D(I) =-( A(I)*UAVE(I) +B(I)*VAVE(I) + AAVE(I))/UCAP
      UABC(I)= A(I)*F(I) - B(I)*E(I)
      C(I)  = (B(I)*M(I) - D(I)*F(I))/UABC(I)
      G(I)  = (D(I)*E(I) - M(I)*A(I))/UABC(I)
      FX(I)  = 1.0/SQRTF(A(I)**2 + B(I)**2 + D(I)**2)
      FY(I)  = 1.0/SQRTF(E(I)**2 + F(I)**2 + M(I)**2)
      FZ(I)  = 1.0/SQRTF( C(I)**2 +  G(I)**2 + 1.0)
      BETA(I,1,1) =  FX(I)*A(I)
      BETA(I,1,2) =  FX(I)*B(I)
      BETA(I,1,3) =  FX(I)*D(I)
      BETA(I,2,1) =  FY(I)*E(I)
      BETA(I,2,2) =  FY(I)*F(I)
      BETA(I,2,3) =  FY(I)*M(I)
      BETA(I,3,1) =  FZ(I)* C(I)
      BETA(I,3,2) =  FZ(I)* G(I)
  210 BETA(I,3,3) =  FZ(I)
C  THE NORMAL DERIVATIVES ARE NOW CALCULATED.
      DO 211 I=1,3
      AARRAY(I,1) = X(5) - X(I)
      AARRAY(I,2) = Y(5) - Y(I)
      AARRAY(I,3) = T(5) - T(I)
      BARRAY(I,1)=U(5) - U(I)
      BARRAY(I,2)=V(5) - V(I)
  211  BARRAY(I,3) = P&PRE - P(I)
      ARGA=1.0
      M=XSIMEQF(3,3,3,AARRAY,BARRAY,ARGA,ARGB)
      GO TO(212,213,214),M
  213 PRINT230
  230  FORMAT (43H OVER/UNDERFLOW NORMAL DERIVATIVES IN BDYPT)
      RETURN
  214 PRINT 231
  231  FORMAT (44H MATRIX SINGULAR NORMAL DERIVATIVES IN BDYPT)
      RETURN
C  THE COMPATIBILITY EQUATIONS ARE SOLVED NEXT.
  212 DO 215 I=1,2
      DO215J=1,3
      DDBETA(I,J) = AARRAY(1,J)*BETA(I,3,1) + AARRAY(2,J)*BETA(I,3,2)
     1     + AARRAY(3,J)*BETA(I,3,3)
  215  VDELBE(I,J)=UAVE(I)*BETA(I,J,1)+VAVE(I)*BETA(I,J,2)
     1     + UCAP*BETA(I,J,3)
      DO216 I=1,2
      DO217 J= 1,2
  217 BARRAY(I,J) = BETA(I,2,J)*VDELBE(I,1) - BETA(I,1,J)*VDELBE(I,2)
      BARRAY(I,3) = (VDELBE(I,1)*VDELBE(I,2)/AAVE(I)**2-BETA(I,2,1)*BE
     1  TA(I,1,1) - BETA(I,2,2)*BETA(I,1,2))/ROEAVE(I)
      E(I) = SQRTF((X(5)-X(I))**2+(Y(5)-Y(I))**2+(T(5)-T(I))**2)
      AARRAY(I,1)=U     (I)*BARRAY(I,1)+V     (I)*BARRAY(I,2)+P     (I)*BA
     1RRAY(I,3)
      F(I) = VDELBE(I,1)*(DDBETA(I,1)*BETA(I,3,1) +DDBETA(I,2)*BETA(I,3,
     12))-VDELBE(I,3)*(DDBETA(I,1)*BETA(I,1,1)+DDBETA(I,2)*BETA(I,1,2))
     2 +DDBETA(I,3)*(VDELBE(I,3)*VDELBE(I,1)/AAVE(I)**2-BETA(I,3,1)
     3*BETA(I,1,1)-BETA(I,3,2)*BETA(I,1,2))/ROEAVE(I)
  216 AARRAY(I,1) = AARRAY(I,1) - E(I)*F(I)
      BARRAY(3,1)=B1
      BARRAY(3,2)=B2
      BARRAY(3,3)=0.0
      AARRAY(3,1)=    -  UCAP*B3
      ARGA=1.0
  235 M = XSIMEQF (3, 3, 1, BARRAY, AARRAY, ARGA,ARGB)
      GO TO (219,220,221),M
  220 PRINT 232
  232  FORMAT (42H OVER/UNDERFLOW COMPATIBILITY EQS IN BDYPT)
      RETURN
  221 PRINT 233
  233  FORMAT (43H MATRIX SINGULAR COMPATIBILITY EQS IN BDYPT)
      RETURN
  219 U(5)=BARRAY(1,1)
      V(5)=BARRAY(2,1)
      P(5)=BARRA (3,1)
C  THE STREAMLINE IS PROJECTED BACK TO DETERMINE THE ENTROPY NEXT.
      U&DUM = (U(5) + U6)/2.0
      V&DUM = (V(5) + V6)/2.0
      AARRAY(1,1)=Y (4)-Y (3)
      AARRAY(1,2)=X (3)-X (4)
```

```
      AARRAY(1,3)=0.0
      AARRAY(2,1)=T (4)-T (3)
      AARRAY(2,2)=0.0
      AARRAY(2,3)=X (3)-X (4)
      AARRAY(3,1)=B3*V4DUM-B2*UCAP
      AARRAY(3,2)=B1*UCAP-B3*U4DUM
      AARRAY(3,3)=B2*U4DUM-B1*V4DUM
      BCOL(1   )=X (3)*(Y (4)-Y (3))-Y (3)*(X (4)-X (3))
      BCOL(2   )=X (3)*(T (4)-T (3))-T (3)*(X (4)-X (3))
      BCOL(3   ) = AARRAY(3,1)*X(5) + AARRAY(3,2)*Y(5) + AARRAY(3,3)*T(5)
      ARGA=1.0
      N=KSIMEQF(3,3,1,AARRAY,BCOL,ARGA,ARGB)
      X6    =AARRAY(1,1)
      Y6    =AARRAY(2,1)
      T6    =AARRAY(3,1)
      AA2=SQRTF((X6    -X (4))**2+(Y6    -Y (4))**2+(T6    -T (4))**2)
      BB2=SQRTF((X (3)-X (4))**2+(Y (3)-Y (4))**2+(T (3)-T (4))**2)
      S(5) =(S (3)-S (4))*AA2/BB2+S (4)
      U6    =(U (3)-U (4))*AA2/BB2+U (4)
      V6    =(V (3)-V (4))*AA2/BB2+V (4)
      IF (P(5)) 300, 301, 301
  300 P(5) = 0.0
  301 ROE4 = ROEDET (P(5), S(5))
      A4 = ADET (P(5), S(5))
C  OUTER LOOP CONVERGENCE TESTS FOLLOW.
      VTOT = SQRTF (U(5)**2 + V(5)**2)
      IF (VTOT - P(5)) 800, 800, 801
  800 IF (ABSF(P(5) - P4PRE) - VTEST*P(5)) 8, 8, 10
  801 IF (ABSF(VTOT - VTOTK) - VTEST*VTOT) 8, 8, 10
C  AVERAGE VALUES ARE CALCULATED BEFORE REPEATING THE ITERATION.
   10 P4PRE = P(5)
      VTOTK = VTOT
      DO 11 I=1,2
      UAVE(I) = (U(I) + U(5))/2.0
      VAVE(I) = (V(I) + V(5))/2.0
      PAVE(I) = (P(I) + P(5))/2.0
      SAVE(I) = (S(I) + S(5))/2.0
      ROEAVE(I) = ROEDET (PAVE(I), SAVE(I))
   11 AAVE(I)=ADET(PAVE(I),SAVE(I))
   21 PRINT 22, X(5) , Y(5), T(5)
   22 FORMAT(44H EXCESS LOOPING OUTER LOOP IN BDYPT, X(4) = , E15.8,
     1 8H Y(4) = , E15.8, 8H T(4) = , E15.8)
    8 IF (P(5)) 302, 302, 303
  302 PRINT 304, X(5), Y(5), T(5), U(5), V(5), P(5), S(5)
  304 FORMAT (/30H NOTE P(5) IN BDYPT AT X(4) = , E15.8, 8H Y(4) = , E15
     1.8, 8H T(4) = , E15.8/ 8H U(4) = , E15.8, 8H V(4) = , E15.8, 8H P(
     24) = , E15.8, 8H S(4) = , E15.8/)
  303 RETURN
      END
```

```
      SUBROUTINE XCHANG(X, Y, T, UCAP, AARRAY, BCOL, B1, B2, B3, B4, DEL
     1TAX, DELTAY, DELTAZ, KDUM, N, RESIDU)
C     XCHANG FOR STEADY CYLINDER WHOSE EQUATION IS B = X**2 + Y**2 - 1.0 = 0
      DIMENSION X(5), Y(5), T(5), AARRAY(3,3), BCOL(3  ), RESIDU(3)
      GO TO (10, 2, 9, 9), KDUM
   10 B1 = (X(3) + X(4))/2.0
      B2 = (Y(3) + Y(4))/2.0
      B3 = 0.0
      B4 = 1.0
      RETURN
    2 IF (N - 1) 4, 3, 4
    3 B1 = 2.0*X(5)
      B2 = 2.0*Y(5)
      B3 = 0.0
      B4 = B1*X(5) + B2*Y(5)
      IF (KDUM - 4) 6, 6, 4
    6 RETURN
    4 AARRAY(3,1) = B1 + DELTAX
      AARRAY(3,2) = B2 + DELTAY
      AARRAY(3,3) = 0.0
      BCOL(3   ) = - (X(5)**2 + Y(5)**2 - 1.0)
      RETURN
    9 RESIDU(3) = 1.0 - 1.0/(X(5)**2 + Y(5)**2)
      RETURN
      END
```

```
      SUBROUTINE XCHANG(X,Y,T,UCAP,A,B,B1,B2,B3,B4,DX,DY,DT,K,N,RES)
      DIMENSION X(5),Y(5),T(5),A(3,3),B(3),RES(3),DUM(2)
C     THIS XCHANG OSCILLATES A CIRCLE VERTICALLY.   THE FIRST FOUR
C     ARITHMETIC STATEMENTS REFER TO THE GEOMETRY AND FREQUENCY.
C  THE EQUATION OF THE SURFACE IS
C  B = X**2 + AA*(Y - BB*(COS(CC*T) - 1.0))**2 - DD = 0.0
      IF (LSTART) 21, 20, 21
   20 READ 22, AA, BB, CC, DD
      PRINT22, AA, BB, CC, DD
   22 FORMAT (4F20.9)
      LSTART = 1
   21 GO TO(1,2,3,4),K
    1 DO 10 J=1,2
   10   DUM(J)=Y(J+2)-BB*(COS(CC*T(J+2))-1.0)
      B1=X(3)+X(4)
      B2=AA*(DUM(1)+DUM(2))
      B3=AA*BB*CC*(DUM(1)*SINF(CC*T(3))+DUM(2)*SINF(CC*T(4)))
      B4=B1*(X(3)+X(4))/2.0+B2*(Y(3)+Y(4))/2.0+B3*(T(3)+T(4))/2.0
      RETURN
    2 IF(N-1)6,4,6
    4 DUM(1)=Y(5)-BB*(COS(CC*T(5))-1.0)
      B1=2.0*X(5)
      B2=2.0*AA*DUM(1)
      B3=2.0*AA*BB*CC*DUM(1)*SINF(CC*T(5))
      B4=B1*X(5)+B2*Y(5)+B3*T(5)
      IF(K-4)6,5,6
    5 RETURN
    6 A(3,1)=2.0*B1+DX*2.0
      A(3,2)=2.0*B2+DY*2.0*AA+DT*2.0*AA*BB*CC*SINF(CC*T(5))
      A(3,3)=2.0*B3+DY*2.0*AA*BB*CC*SINF(CC*T(5))+DT*2.0*AA*CC**2*(
     1BB**2*(2.0*(SINF(CC*T(5)))**2-1.0)+BB*(Y(5)+BB)*COSF(CC*T(5)))
      B(3)=-2.0*(X(5)**2+AA*DUM(1)**2-DD)
      RETURN
    3 RES(3) =  1.0-DD /(X(5)**2    +AA*DUM(1)**2    )
      RETURN
      END
```

```
      SUBROUTINE XCHANG(X,Y,T,UCAP,A,B,B1,B2,B3,B4,DX,DY,DT,K,N,RES)
      DIMENSION X(5),Y(5),T(5),A(3,3),B(3),RES(3)
C  THIS XCHANG DISTORTS ONE ELLIPSE INTO ANOTHER AT AN ANGLE THETA TO THE
C  FREE STREAM.  THE EQUATION OF THE SURFACE FOR T LESS THAN OR EQUAL TO
C  (UCAP*BETA) IS
C  B(X,Y,T) = ((X*COS(THETA))**2-X*Y*SIN(2.*THETA)+(Y*SIN(THETA))**2)/(GAMMA*
C  (0.5*(ALPHA - 1.)*COS(3.14159270*T/(UCAP*BETA) + 1.)) + 0.5*(ALPHA + 1.)))
C  **2 + ((X*SIN(THETA))**2 + X*Y*SIN(2.*THETA) + (Y*COS(THETA))**2)/DELTA**2
C  - 1.0 = 0.0
C  FOR T GREATER THAN (UCAP*BETA)
C  B(X,Y,T) = X**2*((COS(THETA)/(GAMMA*ALPHA))**2 + (SIN(THETA)/DELTA)**2) +
C  X*Y*SIN(2.*THETA)*(1./DELTA**2 - 1./(GAMMA*ALPHA)**2) + Y**2*((SIN(THETA)
C  /(GAMMA*ALPHA))**2 + (COS(THETA)/DELTA)**2) - 1.0 = 0.0
      PHIF(TT) = 0.5*GA*(AL + 1. - (AL - 1.)*COSF(3.14159270*TT/(UCAP*BE)
     1))
      DPHIF(TT) =   GA*3.14159270*(AL-1.)/(2.*UCAP*BE) *SINF(3.14159270*
     1 TT/(UCAP*BE))
      DIDPHF(TT) =  0.5*GA*(AL-1.)*(3.14159270/(UCAP*BE))**2*COSF(3.14159
     1270* TT/(UCAP*BE))
      IF (LSTART)31, 30, 31
 30   READ  32, AL, BE, GA, DE, THETAD
      PRINT 32, AL, BE, GA, DE, THETAD
 32   FORMAT (6F20.9)
      LSTART = 1
      THETA = THETAD*1.7453293E-02
      SITWTH = SINF(2.*THETA)
      SINTH = SINF(THETA)
      COSTH = COSF(THETA)
      SITH2 = SINTH**2
      COTH2 = COSTH**2
 31   GO TO (101,100,100,100),K
 101  IF(T(4)-UCAP*BE  )8,8,9
 100  IF(T(5)-UCAP*BE  )8,8,9
  8   GO TO(1,2,3,4),K
  9   GO TO(21, 22, 23, 24), K
  1   B1=.5*(  2.0*X(3)*(COTH2/PHIF(T(3))**2+SITH2/DE**2)+Y(3)*(SITWTH
     1/DE**2-SITWTH/PHIF(T(3))**2)  +2.0*X(4)*(COTH2/PHIF(T(4))**2
     2+SITH2/DE**2)+Y(4)*(SITWTH/DE**2-SITWTH/PHIF(T(4))**2))
      B2=.5*(2.0*Y(3)*(SITH2/PHIF(T(3))**2+COTH2/DE**2)+X(3)*(SITWT
     1H/DE**2-SITWTH/PHIF(T(3))**2)  +2.0*Y(4)*(SITH2/PHIF(T(4))**2
     2+COTH2/DE**2)+X(4)*(SITWTH/DE**2-SITWTH/PHIF(T(4))**2))
      B3=     -DPHIF(T(3))*(X(3)*COSTH-Y(3)*SINTH)**2/PHIF(T(3))**3
     1-DPHIF(T(4))*(X(4)*COSTH-Y(4)*SINTH)**2/PHIF(T(4))**3
      B4=B1*(X(3)+X(4))/2.0+B2*(Y(3)+Y(4))/2.0+B3*(T(3)+T(4))/2.0
      RETURN
  2   IF(N-1)6,4,6
  4   B1=       2.0*X(5)*(COTH2/PHIF(T(5))**2+SITH2/DE**2)+Y(5)*(SITWTH
     1/DE**2-SITWTH/PHIF(T(5))**2)
      B2=     2.0*Y(5)*(SITH2/PHIF(T(5))**2+COTH2/DE**2)+X(5)*(SITWT
     1H/DE**2-SITWTH/PHIF(T(5))**2)
      B3=-2.0*DPHIF(T(5))*(X(5)*COSTH-Y(5)*SINTH)**2/PHIF(T(5))**3
      B4=B1*X(5)+B2*Y(5)+B3*T(5)
      IF(K    -4)6,5,6
  5   RETURN
  6     A(3,1)=B1+DX*(COTH2/PHIF(T(5))**2 +SITH2/DE**2) +(DY/2.0)*(SITWTH
     1/DE**2-SITWTH/PHIF(T(5))**2)+DT*(-2.0*X(5)*COTH2+Y(5)*SITWTH)*
     2DPHIF(T(5))/PHIF(T(5))**3
      A(3,2)=B2+(DX/2.0)*(SITWTH/DE**2-SITWTH/PHIF(T(5))**2) + DY*
     1(SITH2/PHIF(T(5))**2+COTH2/DE**2)+DT*(-2.0*Y(5)*SITH2+X(5)
     2*SITWTH)*DPHIF(T(5))/PHIF(T(5))**3
      A(3,3)=B3+  DX*(-2.0*X(5)*COTH2+Y(5)*SITWTH)*DPHIF(T(5))/PHIF
     1(T(5))**3+DY*(-2.0*Y(5)*SITH2+X(5)*SITWTH)*DPHIF(T(5))/PHIF
     2(T(5))**3  -DT*(X(5)*COSTH -Y(5)*SINTH)**2*(DIDPHF(T(5))
     3*PHIF(T(5))**3-3.0*DPHIF(T(5))**2)/PHIF(T(5))**4
      B(3)=-X(5)**2*(COTH2/PHIF(T(5))**2+SITH2/DE**2)  -X(5)*Y(5)*
     1(SITWTH/DE**2-SITWTH/PHIF(T(5))**2)  - Y(5)**2*(SITH2/PHIF(T(5))
     2**2+COTH2/DE**2) +1.0
      RETURN
  3   RES(3) = 1. - 1./(X(5)**2*(COTH2/PHIF(T(5))**2 + SITH2/DE**2)
     1 + X(5)*Y(5)*(SITWTH/DE**2 - SITWTH/PHIF(T(5))**2) +Y(5)**2*(SITH2
     2/PHIF(T(5))**2 + COTH2/DE**2))
      RETURN
 21   B1=(X(3)+X(4))*(COTH2/(GA*AL)**2+SITH2/DE**2)  +.5*(Y(3)+Y(4))*
     1(SITWTH/DE**2-SITWTH/(GA*AL)**2)
      B2=.5*(X(3)+X(4))*(SITWTH/DE**2-SITWTH/(GA*AL)**2)  +(Y(3)+Y(4))*
     1(SITH2/(GA*AL)**2+COTH2/DE**2)
      B3=0.0
      B4=B1*(X(3)+X(4))/2.0+B2*(Y(3)+Y(4))/2.0
      RETURN
 22   IF(N-1)26,24,26
 24   B1=2.0*X(5)*(COTH2/(GA*AL)**2+SITH2/DE**2)+Y(5)*(SITWTH/DE**2
     1-SITWTH/(GA*AL)**2)
      B2=X(5)*(SITWTH/DE**2-SITWTH/(GA*AL)**2)  +2.0*Y(5)*
     1(SITH2/(GA*AL)**2+COTH2/DE**2)
      B3=0.0
      B4=B1*X(5)+B2*Y(5)
      IF(K-4)26,25,26
 25   RETURN
 26   A(3,1)=B1+DX*(COTH2/(GA*AL)**2+SITH2/DE**2) + (DY/2.0)*(SITWTH
     1/DE**2-SITWTH/(GA*AL)**2)
      A(3,2)=B2+(DX/2.0)*(SITWTH/DE**2-SITWTH/(GA*AL)**2)
     1+DY*(SITH2/(GA*AL)**2+COTH2/DE**2)
      A(3,3)=0.0
      B(3)=-(X(5))**2*(COTH2/(GA*AL)**2+SITH2/DE**2) -X(5)*Y(5)*(SITWT
     1H/DE**2-SITWTH/(GA*AL)**2)-Y(5)**2*(SITH2/(GA*AL)**2+
     2COTH2/DE**2)+ 1.0
      RETURN
 23   RES(3) = 1. - 1./(X(5)**2*(COTH2/(GA*AL)**2 + SITH2/DE**2)
     1 + X(5)*Y(5)*(SITWTH/DE**2 -  SITWTH/(GA*AL)**2) + Y(5)**2*(
     2SITH2/(GA*AL)**2 + COTH2/DE**2))
      RETURN
      END
```

```
C  GENERALIZED XCHANG SUBROUTINE
C  THE QUANTITIES IN BOXES ARE TO BE DETERMINED ANALYTICALLY AND WRITTEN IN
C  FORTRAN.  THE EQUATION OF THE SURFACE IS ASSUMED TO HAVE THE FORM
C
C                              B(x, y, t') = 0
C
      SUBROUTINE XCHANG (X, Y, T, U, A, B, B1, B2, B3, B4, DX, DY, DT, K
     1, N, RES)
      DIMENSION X(5), Y(5), T(5), A(3,3), B(3), RES(3)
      IF (LSTART) 10, 11, 10

 11   READ 12, | LIST OF GEOMETRIC PARAMETERS
      PRINT 12, | WHICH DEFINE THE SURFACE.

 12   FORMAT (6F20.9)
      LSTART = 1
 10   GO TO (1, 2, 3, 4), K
```

```
1     B1 = [ ∂B/∂x (x3, y3, t'3) + ∂B/∂x (x4, y4, t'4) ] /2.0
      B2 = [ ∂B/∂y (x3, y3, t'3) + ∂B/∂y (x4, y4, t'4) ] /2.0
      B3 = [ ∂B/∂t' (x3, y3, t'3) + ∂B/∂t' (x4, y4, t'4) ] /2.0
      B4 = B1*(X(3) + X(4))/2.0 + B2*(Y(3) + Y(4))/2.0 + B3*(T(3) + T(4)
     1)/2.0
      RETURN
2     IF (N-1) 4, 3, 3
4     B1 = ∂B/∂x (x3, y3, t'3)
      B2 = ∂B/∂y (x3, y3, t'3)
      B3 = ∂B/∂t' (x3, y3, t'3)
      B4 = B1*X(3) + B2*Y(3) + B3*T(3)
      IF (N-4) 6, 5, 6
5     RETURN
6     A(3,1) = ∂²B/∂x² (x3, y3, t'3) + (DX/2.0)* ∂³B/∂x³ (x3, y3, t'3)
     1 + (DY/2.0)* ∂³B/∂x²∂y (x3, y3, t'3) + (DT/2.0)* ∂³B/∂x²∂t' (x3, y3, t'3)
      A(3,2) = ∂²B/∂x∂y (x3, y3, t'3) + (DX/2.0)* ∂³B/∂x²∂y (x3, y3, t'3)
     1 + (DY/2.0)* ∂³B/∂x∂y² (x3, y3, t'3) + (DT/2.0)* ∂³B/∂x∂y∂t' (x3, y3, t'3)
      A(3,3) = ∂²B/∂x∂t' (x3, y3, t'3) + (DX/2.0)* ∂³B/∂x²∂t' (x3, y3, t'3)
     1 + (DY/2.0)* ∂³B/∂x∂y∂t' (x3, y3, t'3) + (DT/2.0)* ∂³B/∂x∂t'² (x3, y3, t'3)
      B(3) = - B(x3, y3, t'3)
      RETURN
3     RES(3) = NORMALIZED B(3). BUT DO NOT USE THE FORTRAN TERM B(3) IN
     THE STATEMENT.
      RETURN
      END
C  NOTE THAT THE EQUATIONS FOR B4 MIGHT DEGENERATE IF B1, B2, OR B3 IS ZERO OR
C  IF TERMS CANCEL.  THEN THESE STATEMENTS CAN BE SIMPLIFIED.



      SUBROUTINE XINTER(X,Y,T,U,V,P,S)
C  THE COORDINATES OF THE BASE POINTS ARE INPUT IN THE FIRST THREE LOCATIONS OF
C  X, Y AND T. THE COORDINATES OF THE POINT AT WHICH THE FUNCTION IS TO BE
C  DETERMINED ARE INPUT IN THE FIFTH LOCATION. FOUR FLOW PROPERTIES ARE
C  INTERPOLATED SIMULTANEOUSLY WITH THE RESULTS OUTPUT IN THE FIFTH LOCATIONS OF
C  U, V, P AND S.
C     XINTER REQUIRES ACCESS TO FUNCTION THETA
      DIMENSION X(5),Y(5),T(5),U(5),V(5),P(5),S(5),C(2),XP(3),YP(3)
      EQUIVALENCE (YP(1),THE)
  700 C(1) = X(2) - X(1)
      C(2)=T(2)-T(1)
      THE=THETA(C(1),C(2))
      SINTHE=SINF(THE)
      COSTHE =COSF(THE)
      C(1)=(X(2)-X(1))*COSTHE+(T(2)-T(1))*SINTHE
      C(2)=Y(2)-Y(1)
      PHI=THETA(C(1),C(2))
      SINPHI=SINF(PHI)
      COSPHI=COSF(PHI)
      C(1)= (Y(3)-Y(1))*COSPHI-((X(3)-X(1))*COSTHE+(T(3)-T(1))*
     1SINTHE)*SINPHI
      C(2)=(T(3)-T(1))*COSTHE-(X(3)-X(1))*SINTHE
      PSI=THETA(C(1),C(2))
      SINPSI = SINF (PSI)
      COSPSI = COSF (PSI)
      DO 1 I=1,2
    1 XP(I)=((X(I+1)-X(1))*COSTHE+(T(I+1)-T(1))*SINTHE)*COSPHI
     1+(Y(I+1)-Y(1))*SINPHI
      XP(3)=((X(5  )-X(1))*COSTHE+(T(5  )-T(1))*SINTHE)*COSPHI
     1+(Y(5  )-Y(1))*SINPHI
       YP(2)=((Y(3)-Y(1))*COSPHI-((X(3)-X(1))*COSTHE+(T(3)-T(1))
     1*SINTHE)*SINPHI)*COSPSI+((T(3)-T(1))*COSTHE-(X(3)-X(1))*SINTHE)*SINPSI
       YP(3)=((Y(5)-Y(1))*COSPHI-((X(5)-X(1))*COSTHE+(T(5)-T(1))
     1*SINTHE)*SINPHI)*COSPSI+((T(5)-T(1))*COSTHE-(X(5)-X(1))*SINTHE)*SINPSI
      U(5) = (U(2) - U(1))*XP(3)/XP(1) + YP(3)*((U(3) - U(1))/YP(2)
     1-(U(2)-U(1))*XP(2)/XP(1)*YP(2))+U(1)
      V(5) = (V(2) - V(1))*XP(3)/XP(1) + YP(3)*((V(3) - V(1))/YP(2)
     1-(V(2)-V(1))*XP(2)/XP(1)*YP(2))+V(1)
      P(5) = (P(2) - P(1))*XP(3)/XP(1) + YP(3)*((P(3) - P(1))/YP(2)
     1-(P(2)-P(1))*XP(2)/XP(1)*YP(2))+P(1)
      S(5) = (S(2) - S(1))*XP(3)/XP(1) + YP(3)*((S(3) - S(1))/YP(2)
     1-(S(2)-S(1))*XP(2)/XP(1)*YP(2))+S(1)
      RETURN
      END



      FUNCTION ADET (P,S)
C  GIVEN THE PRESSURE, P, IN LB./SQ.FT. AND THE ENTROPY, S, IN BTU/LBM,DEGREE R,
C  THIS FUNCTION DETERMINES THE SPEED OF SOUND IN FT./SEC.
      ADET = SQRTF(1186469.2*(EXPF(-41.11*S/0.06886)*P/2117.0)
     1**0.28571429)
      RETURN
      END



      FUNCTION ROEDET (P,S)
C  GIVEN THE PRESSURE, P, IN LB./SQ.FT. AND THE ENTROPY, S, IN BTU/LBM,DEGREE R,
C  THIS FUNCTION DETERMINES THE DENSITY IN SLUG/CU.FT.
      ROEDET = 0.002498*(P/2117.0)**0.71428571*EXPF(41.11-S/0.06886)
```

```
     1*0.28571429)
      RETURN
      END



      FUNCTION THETA(X,Y)
C  THIS FUNCTION DETERMINES AN ANGLE IN RADIANS, GIVEN THE OPPOSITE AND ADJACENT
C  SIDES OF A RIGHT TRIANGLE.
C       THETA BETWEEN 0.0 AND 2 PIE
        IF(Y) 10,20,30
 10     THETA=ATANF(-X/Y)+4.712388975
        RETURN
 20     IF (X) 21,22,22
 21     THETA=3.14159265
        RETURN
 22     THETA=0.0
        RETURN
 30     THETA=ATANF(-X/Y)+1.57079633
        RETURN
        END
```

TABLE I

Results of a Typical Field Point Calculation

Input Points (x, y, t' in ft., u, v in fps, p in psf, s in BTU/lbm. ^{o}R)

	x	y	t' (U=2000 fps)
1st point	-1.1447676	0.58913501	0.14216991
2nd point	-1.0939953	0.54756027	0.14148745
3rd point	-1.1141122	0.49530905	0.14097986

	u	v	p	s
1st point	874.92389	897.04218	6071.4579	3.1160775
2nd point	727.29974	923.78306	6218.8732	3.1168170
3rd point	648.66292	849.49358	6405.2653	3.1169245

Convergence Test Values

PATCH	Inner Loop	Middle Loop	Outer Loop
0.0001	0.0001	0.0000005	0.0001

Coordinates of Points Output from PATCH

x	y	t'
-1.1028552	0.55481524	0.14160654
-1.0981097	0.53687342	0.14138363
-1.1279440	0.53764377	0.14151682

First Estimate of the Coordinates

x	y	t'
-1.1079344	0.5487763	0.1552792

First Inner Loop Iteration

Δx	Δy	Δt'
$-1.6167161 \times 10^{-4}$	4.5400817×10^{-4}	1.1548015×10^{-4}
$-1.5992165 \times 10^{-4}$	4.5651031×10^{-4}	1.2058911×10^{-4}
$-1.5992875 \times 10^{-4}$	4.5650098×10^{-4}	1.2056934×10^{-4}
$-1.5992875 \times 10^{-4}$	4.5650102×10^{-4}	1.2056942×10^{-4}

First Middle Loop Result (no iteration required)

x	y	t'
-1.1080943	0.5492328	0.1553947

(Residuals all less than 5×10^{-8})

First Outer Loop Result

u	v	p	s	$\sqrt{u^2 + v^2}$
726.8993	912.92059	6171.1549	3.1168188	1166.9651

Second Inner Loop Iteration

Δx	Δy	Δt'
7.2747344×10^{-5}	1.2188992×10^{-4}	$-5.2904863 \times 10^{-5}$
7.2885107×10^{-5}	1.2169935×10^{-4}	$-5.2511849 \times 10^{-5}$
7.2885356×10^{-5}	1.2169902×10^{-4}	$-5.2511145 \times 10^{-5}$

Second Middle Loop Result

x	y	t'
-1.1080214	0.5491111	0.1553422

Second Outer Loop Result

u	v	p	s	$\sqrt{u^2 + v^2}$
725.35434	915.54065	6175.2055	3.1168192	1168.0555

Third Inner Loop Iteration

Δx	Δy	Δt'
$-5.2475364 \times 10^{-6}$	8.5396863×10^{-6}	$-1.3029714 \times 10^{-6}$
$-5.2465308 \times 10^{-6}$	8.5410568×10^{-6}	$-1.3001144 \times 10^{-6}$
$-5.2465308 \times 10^{-6}$	8.5410568×10^{-6}	$-1.3001143 \times 10^{-6}$

Third Middle Loop Result

x	y	t'
-1.1080266	0.5491196	0.1553409

Third Outer Loop Result

u	v	p	s	$\sqrt{u^2 + v^2}$
725.26143	915.86852	6176.1796	3.1168191	1168.2548

Fourth Inner Loop Iteration

Δx	Δy	$\Delta t'$
$-6.2957056 \times 10^{-7}$	1.5980978×10^{-6}	2.4694767×10^{-7}
$-6.2954510 \times 10^{-7}$	1.5981327×10^{-6}	2.4702034×10^{-7}
$-6.2954509 \times 10^{-7}$	1.5981327×10^{-6}	2.4702034×10^{-7}

Fourth Middle Loop Result

x	y	t'
-1.1080272	0.5491219	0.1553411

Fourth Outer Loop Results

u	v	p	s	$\sqrt{u^2 + v^2}$
725.24378	915.91642	6176.2986	3.1168191	1168.2814

The same point was re-calculated with the following Convergence Test Values.

PATCH	Inner Loop	Middle Loop	Outer Loop
0.00001	0.00001	0.0000001	0.000001

The coordinates from PATCH and the first four iterations through the outer loop were exactly the same as given above. Due to the decrease of the outer loop convergence test value by a factor of 100, the outer loop was iterated two more cycles with the following results.

Fifth Outer Loop Results

x	y	t'
-1.1080286	0.5491215	0.1553407

u	v	p	s	$\sqrt{u^2 + v^2}$
725.24043	915.92667	6176.3179	3.1168191	1168.2874

Sixth Outer Loop Results

x	y	t'
-1.1080277	0.5491208	0.1553404

u	v	p	s	$\sqrt{u^2 + v^2}$
725.23881	915.92730	6176.3199	3.1168191	1168.2868

TABLE II

TIME COORDINATES OF THE TIME SURFACES FOR THE FOUR BLUNT BODY CASES

$t = t'/U = t'/2000$, t in seconds and t' in feet

An asterisk indicates a time surface on which ADDROW was used to add points by interpolation

Time Surface Number	Steady Flow t'	Symmetric Warping t'	Asymmetric Warping t'	Vertical Oscillation t'
0	.0	.0	.0	.0
2	.0858 - .0182	.0858 - .0179	.0858 - .0179	.0858 - .0182
4	.0455	.0426	.0431	.04520
6	.1268 - .0645	.1237 - .0603	.1239 - .0609	.1273 - .0642
8	.0930*	.0838*	0857	.0912*
10	.1650 - .1076	.1555 - .0980	.1568 - .1001	.1652 - .1059
12	.1339	.1202	.1227	.1327
14	.2033 - .1493	.1884 - .1355	.1899 - .1381	.2045 - .1483
16	.1756*	.1606*	.1637*	.1744
18	.2406 - .1888	.2235 - .1744	.2253 - .1774	.2425 - .1905
20	.2153	.2001	.2028	.2174*
22	.2372	.2606 - .2145	.2622 - .2172	.2802 - .2309
24	.2980 - .2513	.2387*	.2415*	.2569
26	.2773*	.2956 - .2507	.2972 - .2539	.3178 - .2709
28	.3364 - .2896	.2748	.3519 - .2689	.2968*
30	.3156	.3301 - . 2872	.2955*	.3554 - .3085
32	.3735 - .3284	.3108	.3489 - .3065	.3359
34	.3542	.3647 - .3236	.3303	.3940 - .3483

TABLE II (continued)

36	.3759*	.3469*	.3824 - .3416	.3748
38	.4304 - .3867	.3986 - .3573	.3650	.4321 - .3880
40	.4124	.3808	.4165 - .3767	.4140*
42	.4341	.4318 - .3915	.3997	.3690 - .4243
44	.4556*	.4146	.4504 - .4116	.4494
46	.5091 - .4648	.4646 - .4255	.4344*	.5035 - .4601
48	.4904	.4482	.4824 - .4437	.4853
50	.5487 - .5000		.4663	.5390 - .4960
52				.5214
54				.5758 - .5321

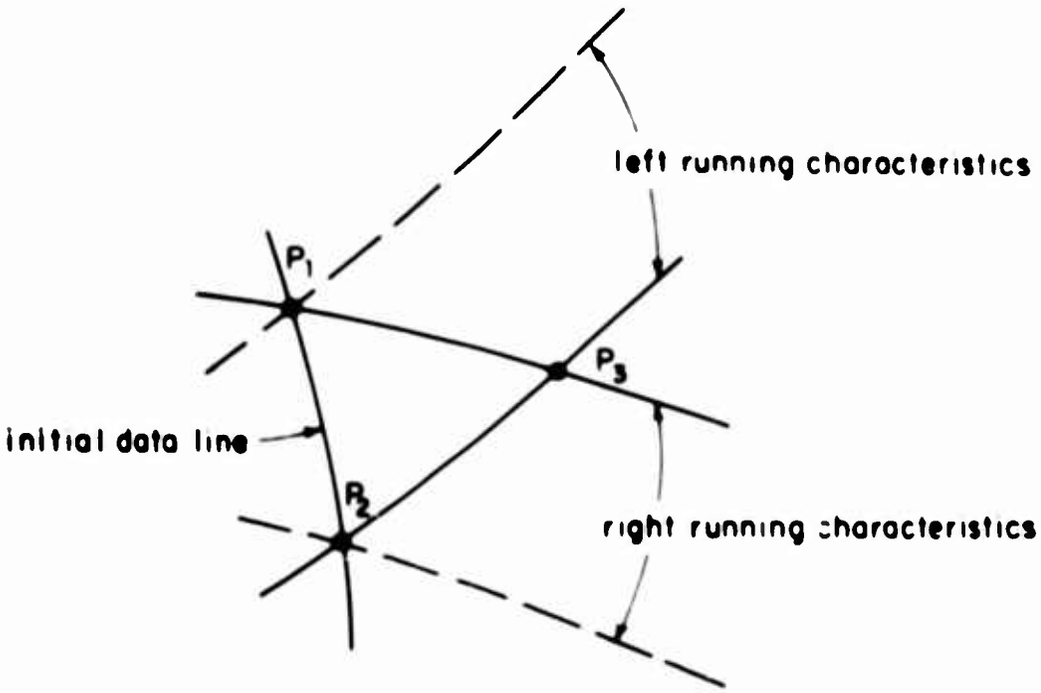


Figure 1 Characteristic net for two independent variables.

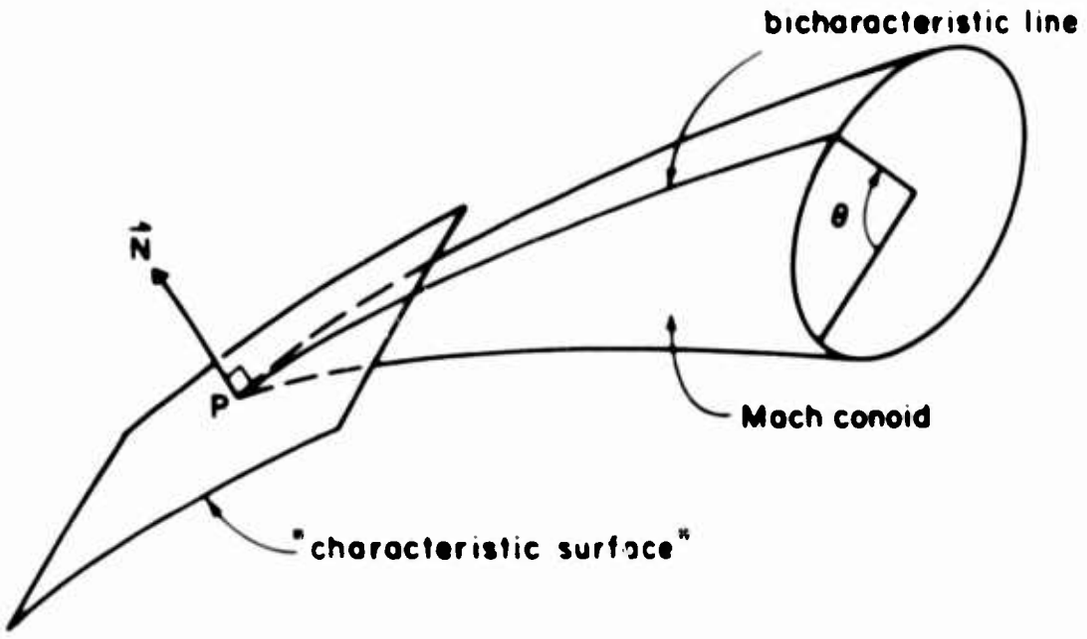


Figure 2 Characteristic conoid for three independent variables.

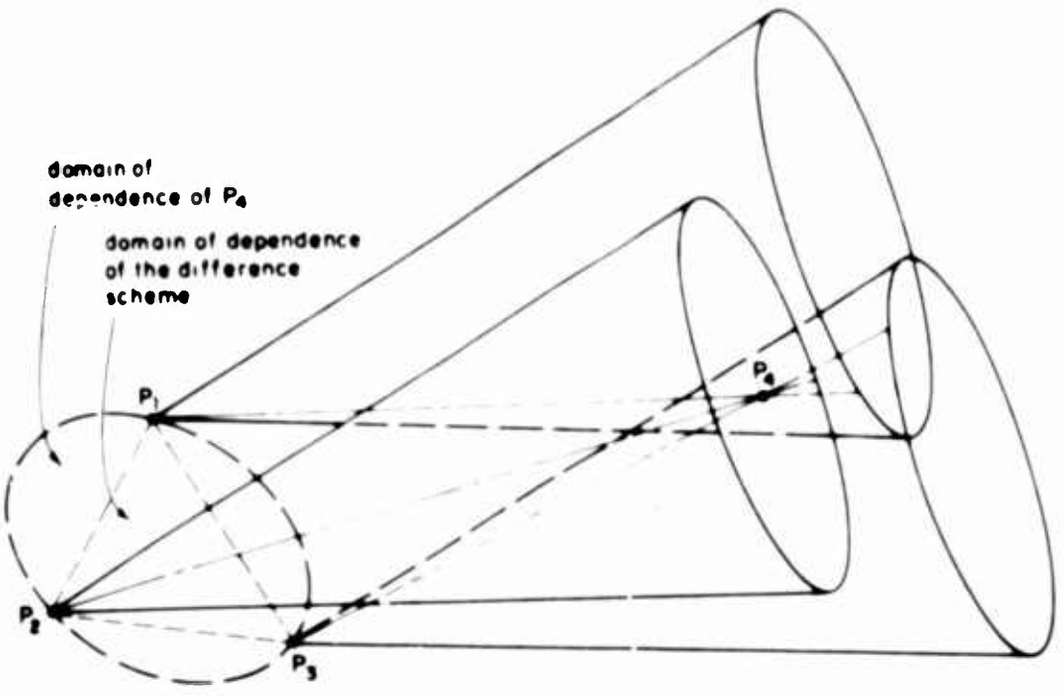


Figure 5 Tetrahedral Characteristic Line Network

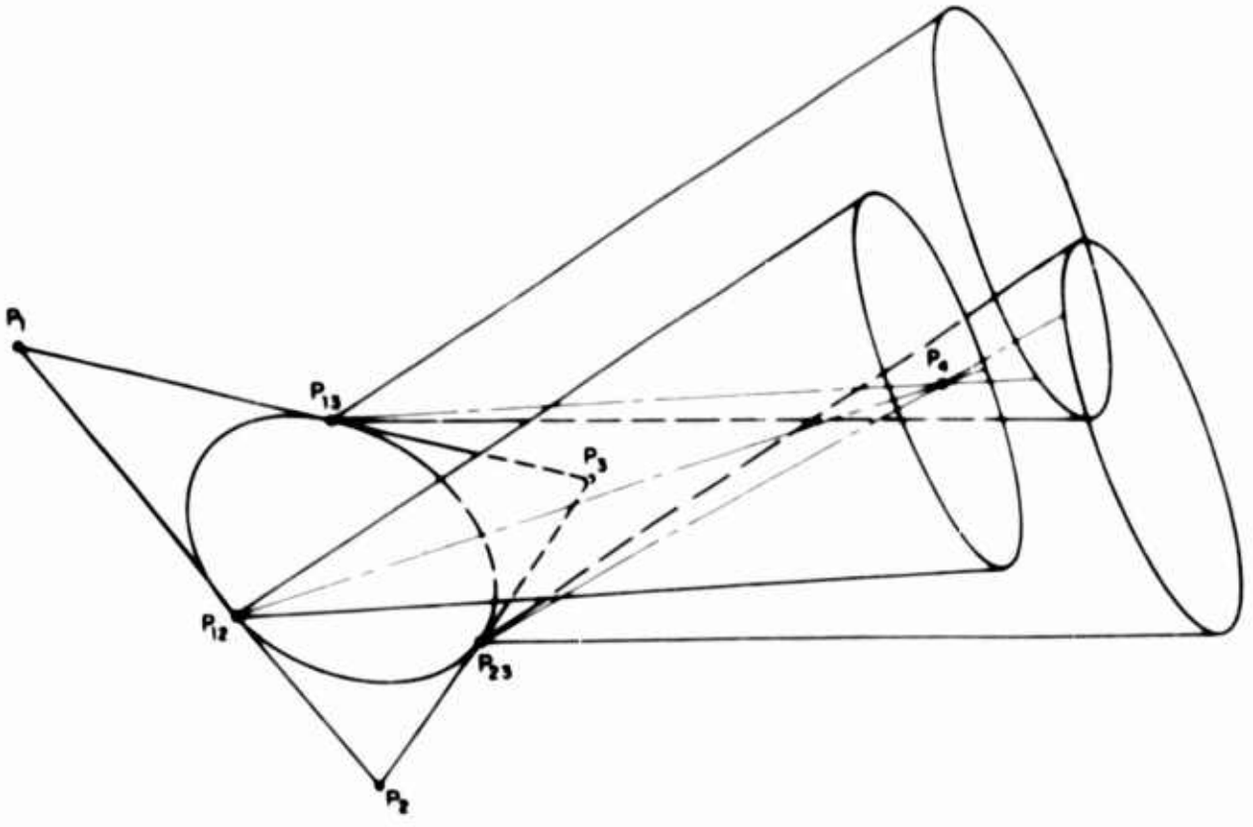


Figure 6 Modified Tetrahedral Characteristic Line Network

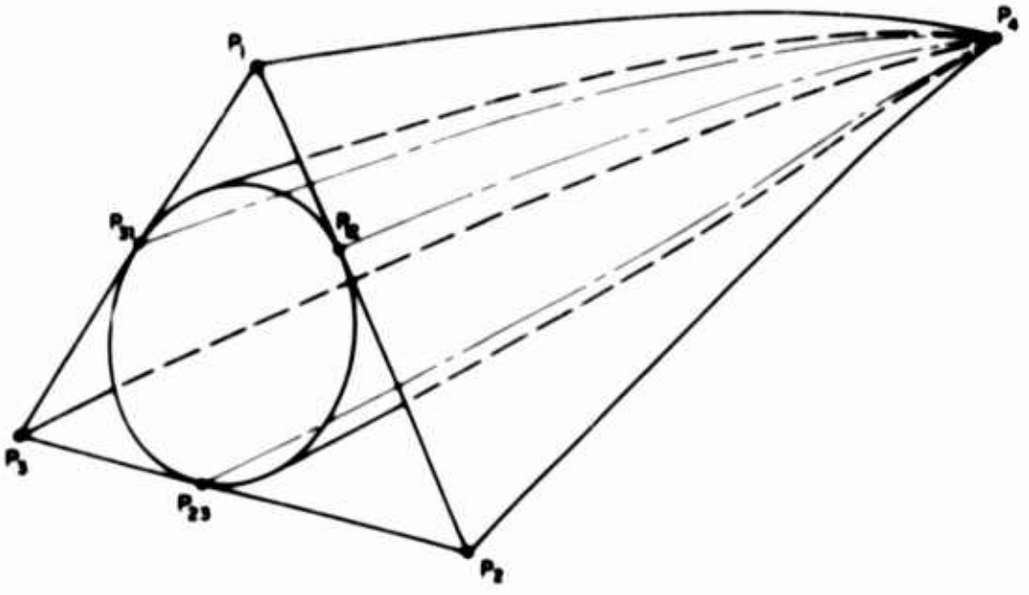


Figure 7 Tetrahedral characteristic surface network

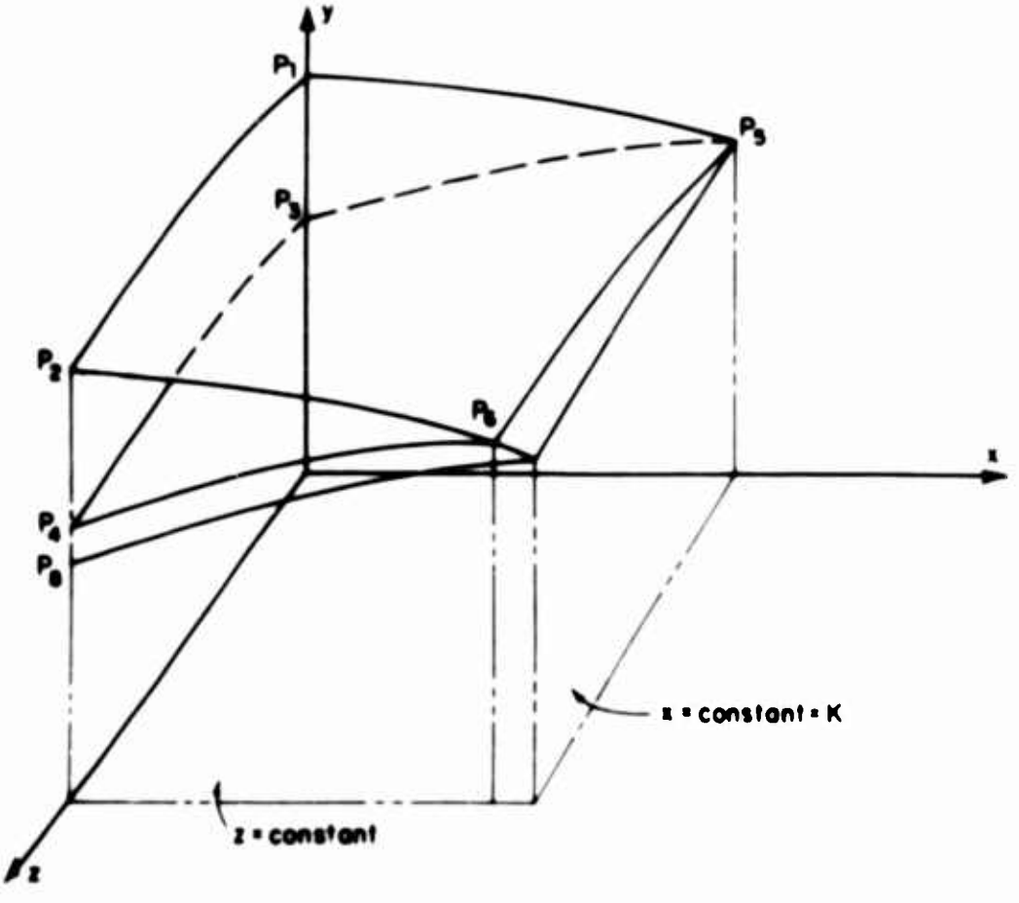


Figure 8 Network of intersection of reference planes with characteristic surfaces.

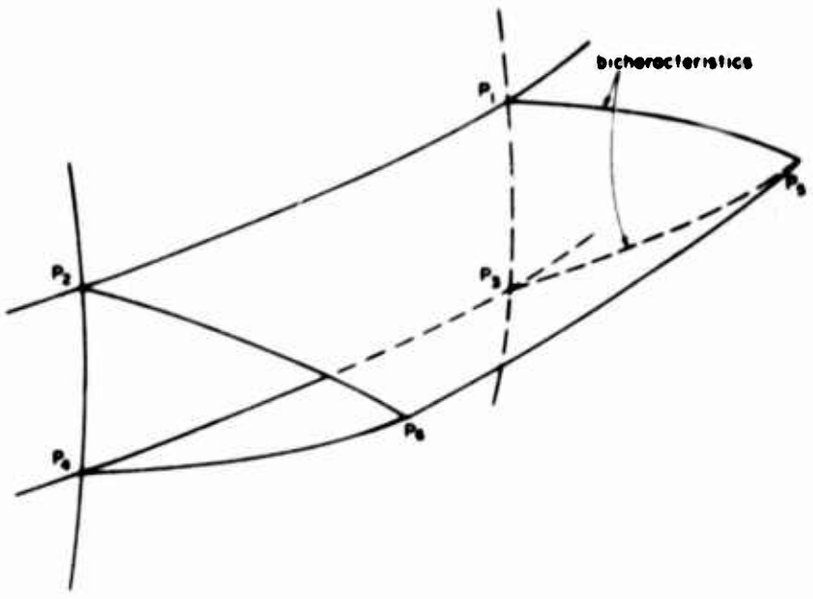


Figure 9 Prismatic Network of Characteristic Surfaces

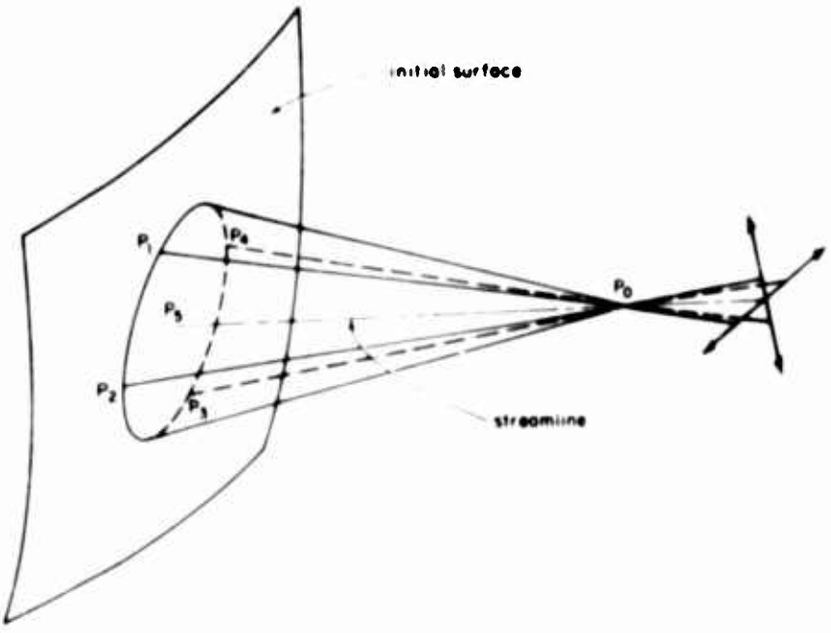


Figure 10 Pentahedral Bicharacteristic Line Network

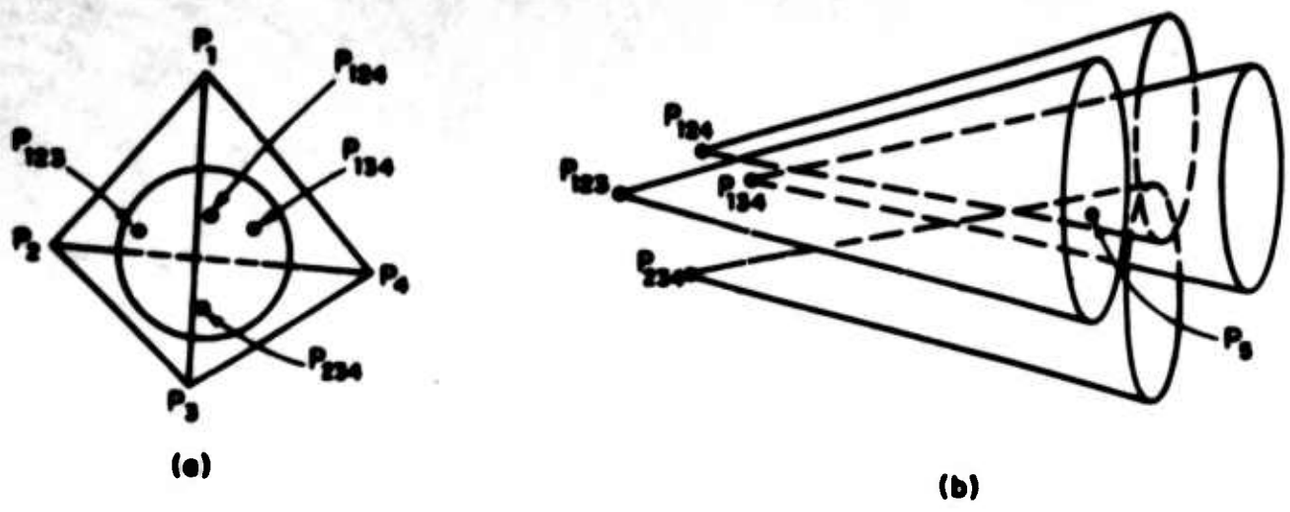


Figure 11 Representation of the modified tetrahedral characteristic line network for four independent variables.

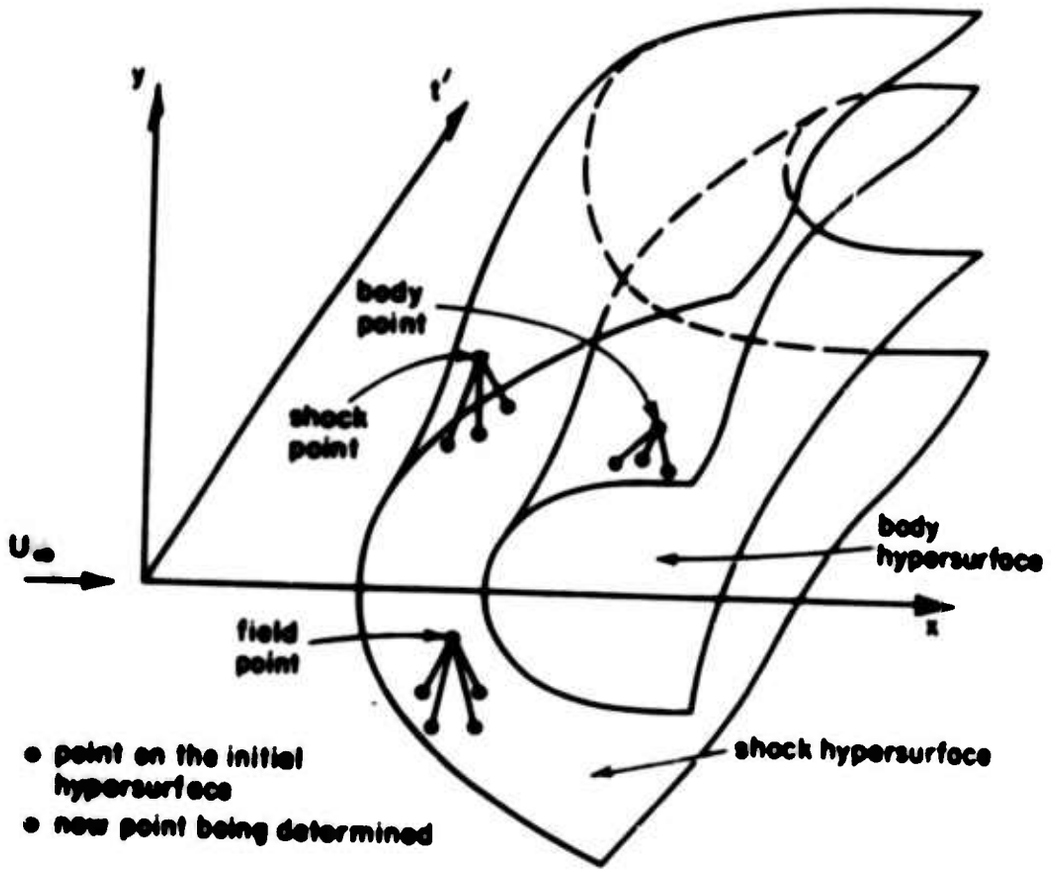


Figure 12 Representation of the basic elements of the numerical integration procedure.

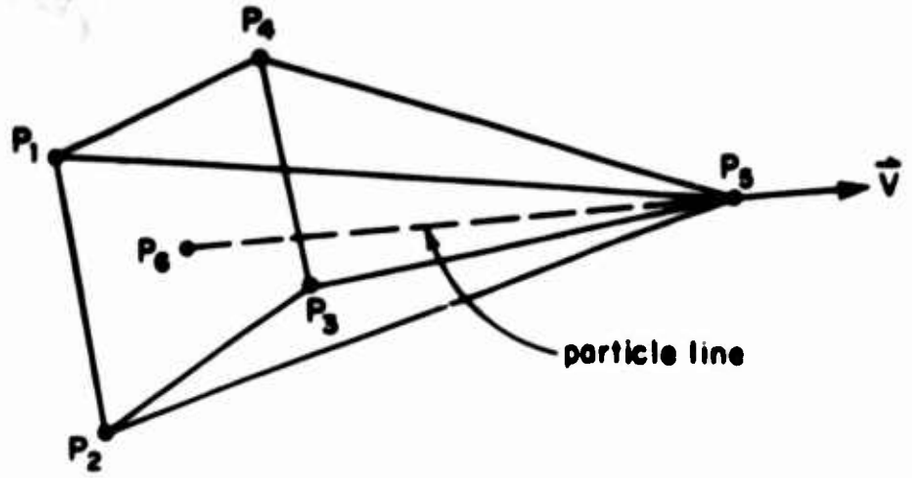


Figure 13 Representation of a portion of the field point network.

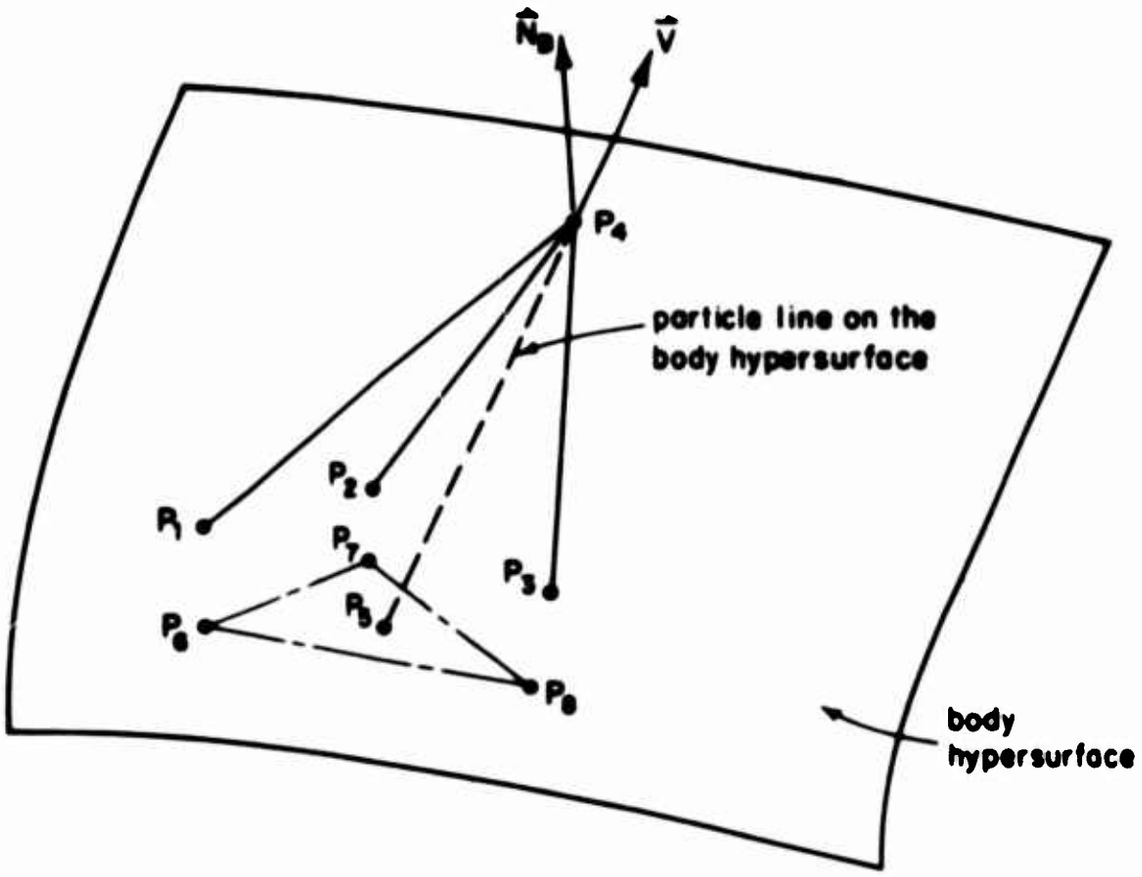


Figure 14 Representation of a portion of the body point network.

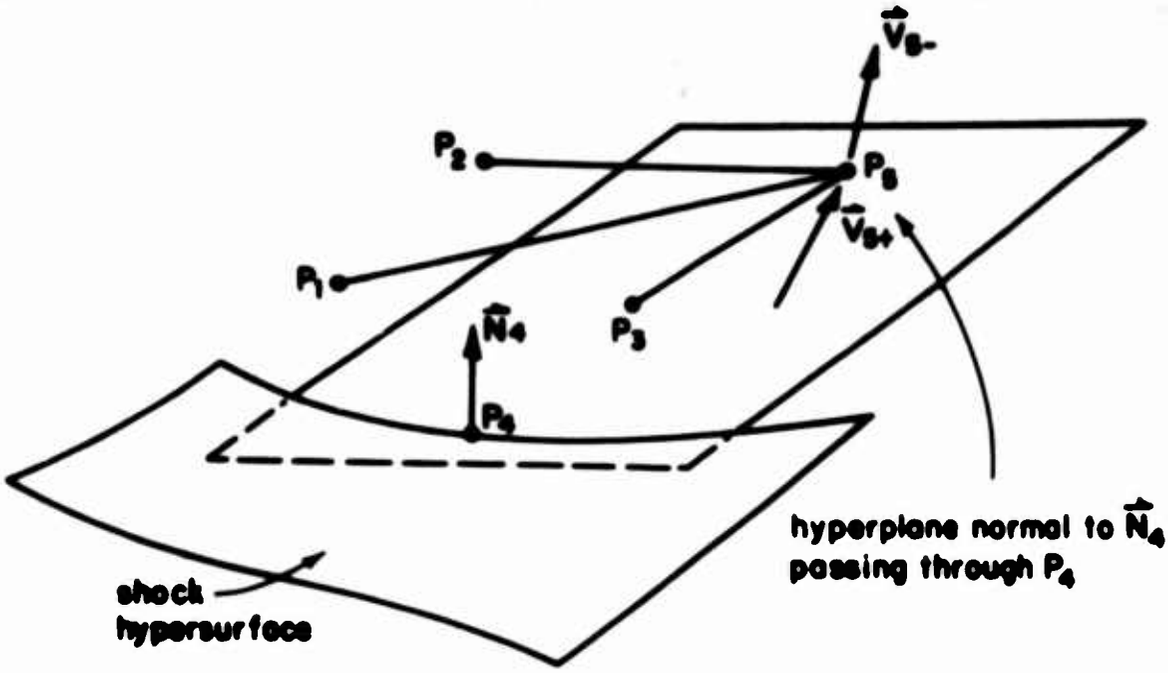


Figure 15 Representation of a portion of the shock point network.

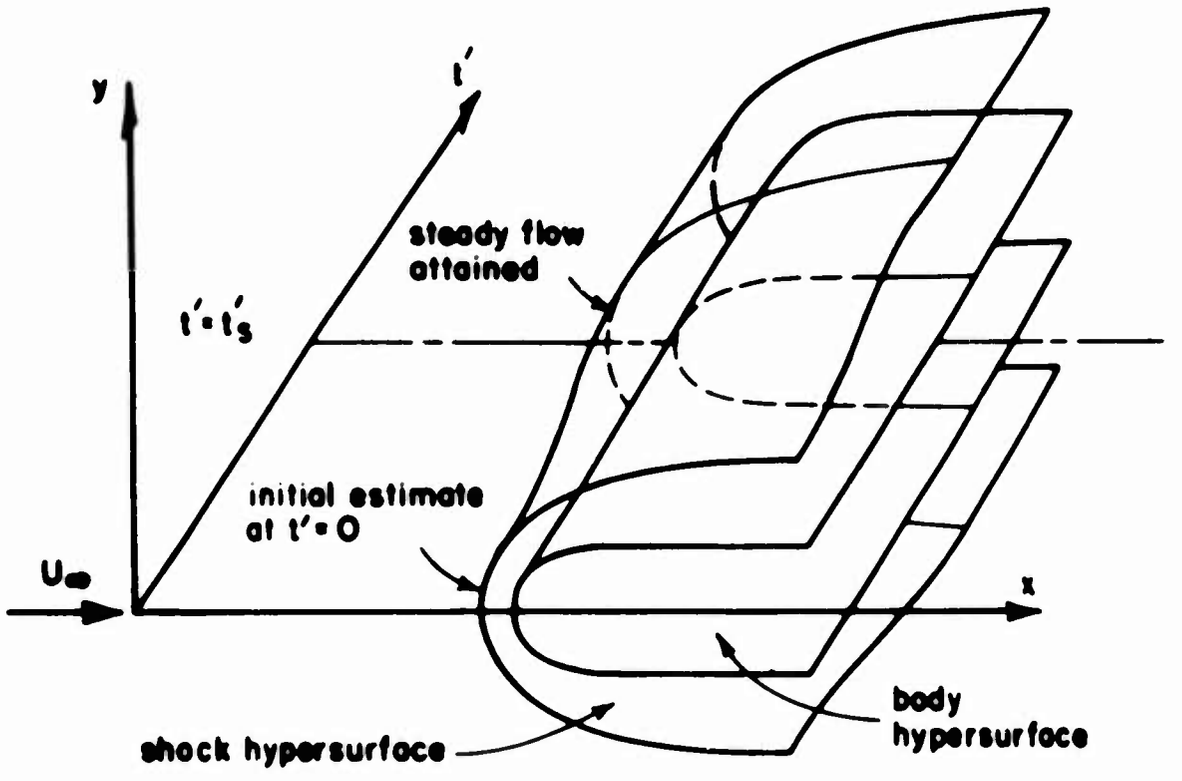


Figure 16 Representation of a method for determining steady flows.

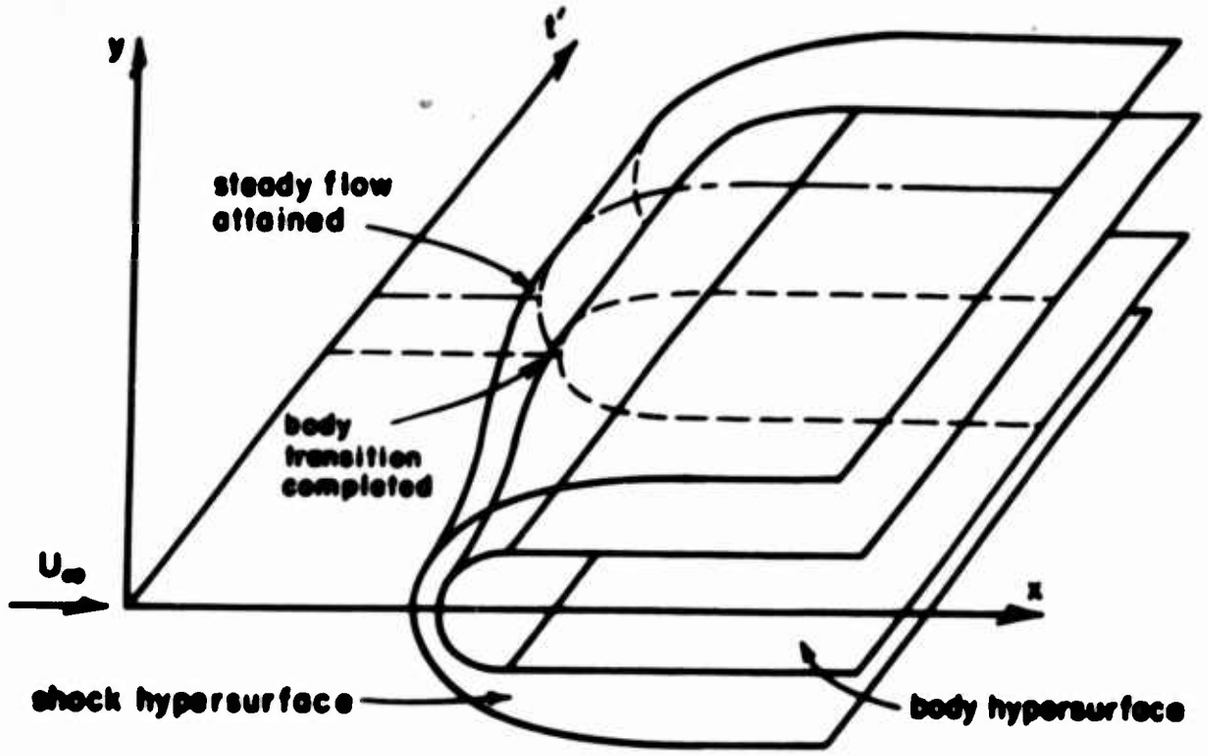


Figure 17 Representation of a second method for determining steady flows.

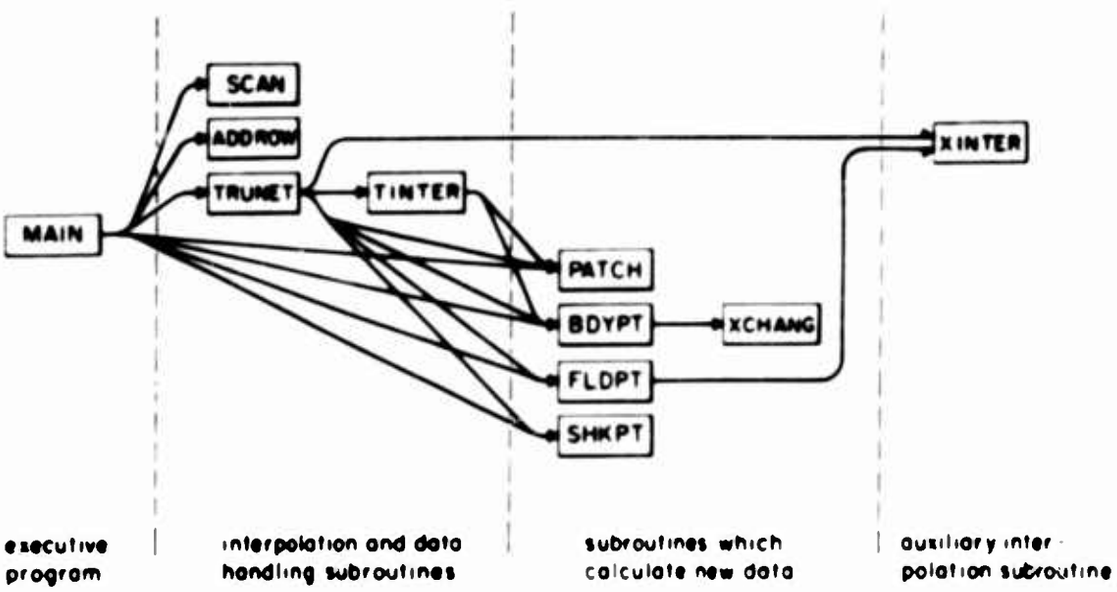
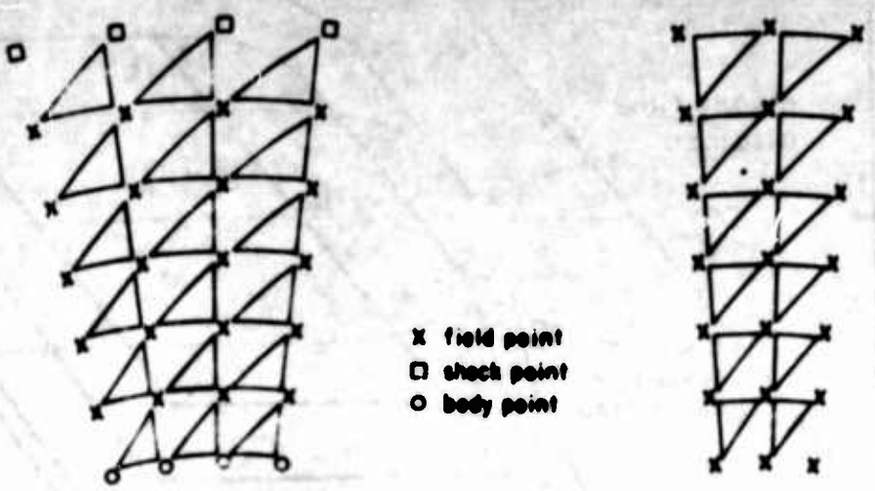


Figure 18 Subroutine calling sequence



even numbered surface showing organization to calculate next odd numbered surface

odd numbered surface showing organization to calculate next even numbered surface

The three points at the vertices of each of the triangles are used to calculate a new point on the next surface.

Figure 19 Executive program organization.

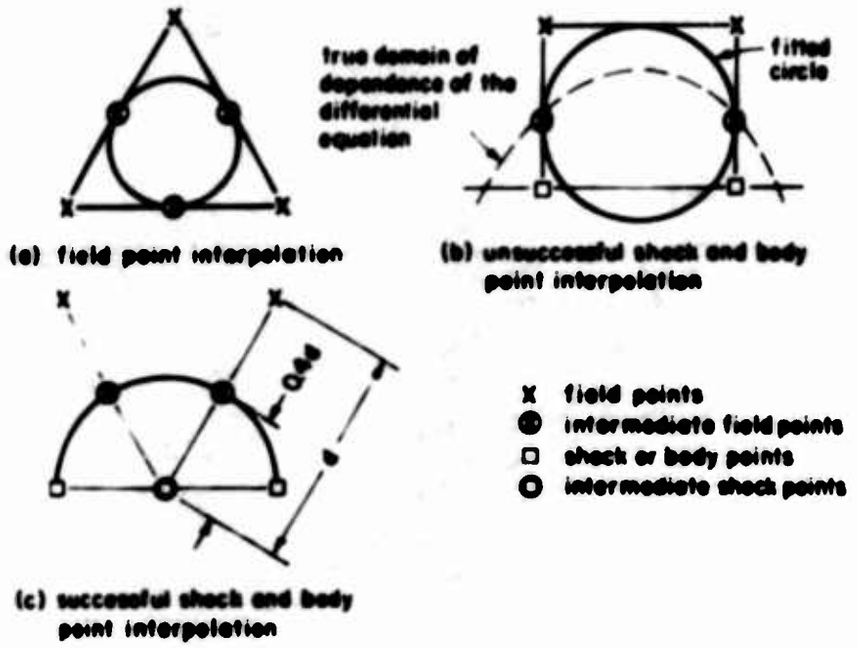


Figure 20 Interpolation schemes used by PATCH in the initial surface.

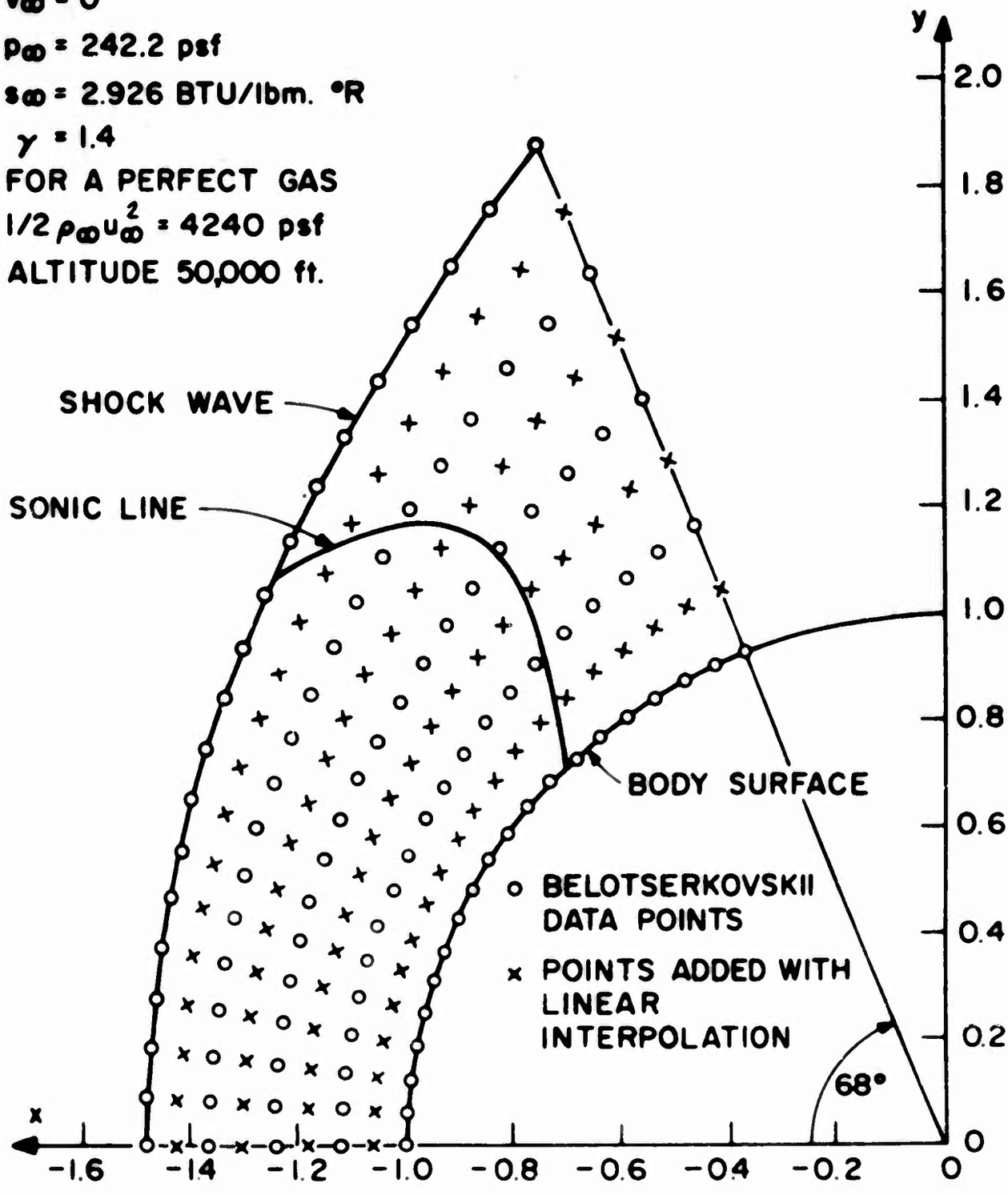


FIGURE 21 BLUNT BODY INITIAL DATA NET

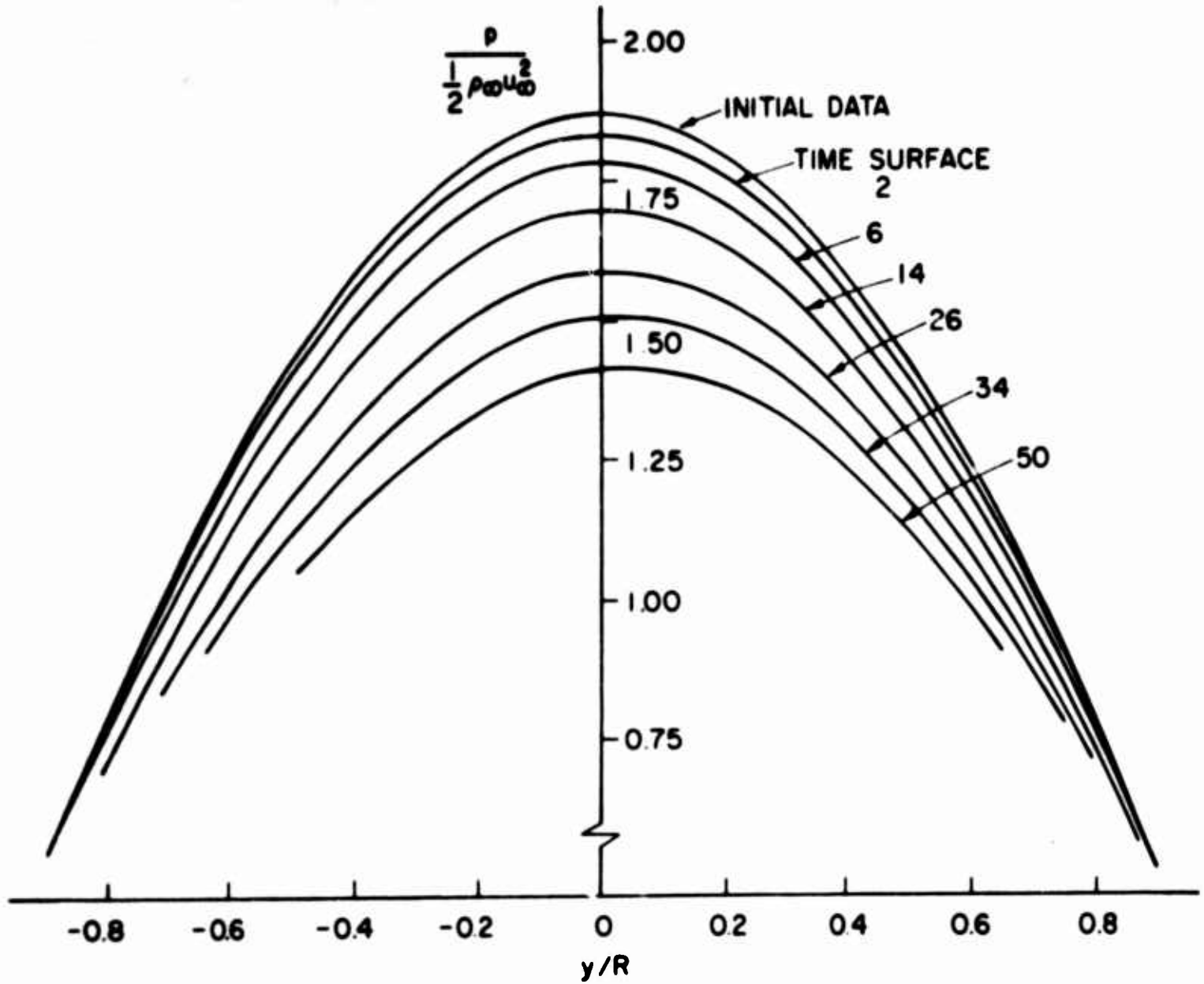


FIGURE 22 BODY SURFACE PRESSURE FOR STEADY FLOW

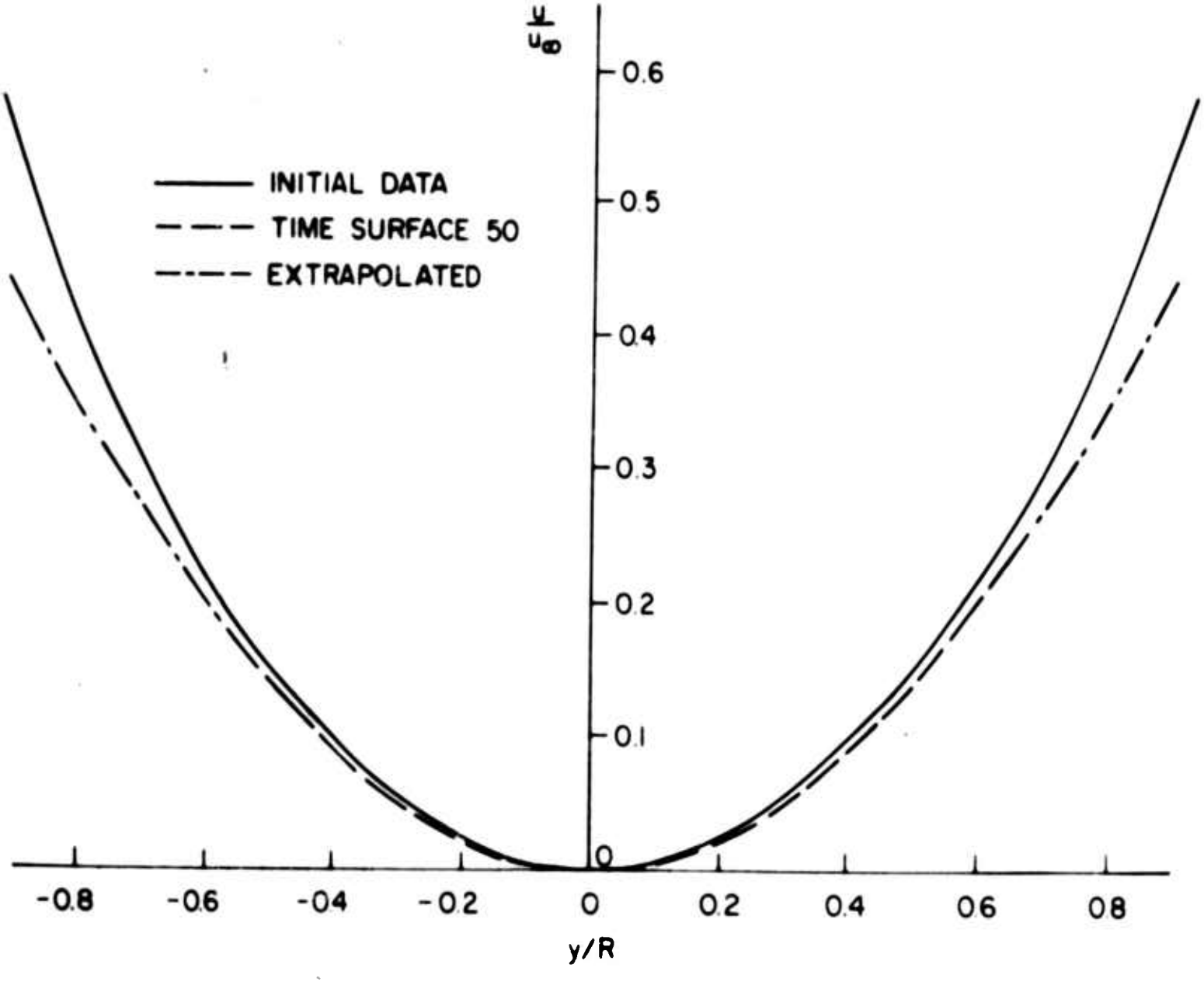


FIGURE 23 BODY SURFACE u VELOCITY COMPONENT FOR STEADY FLOW

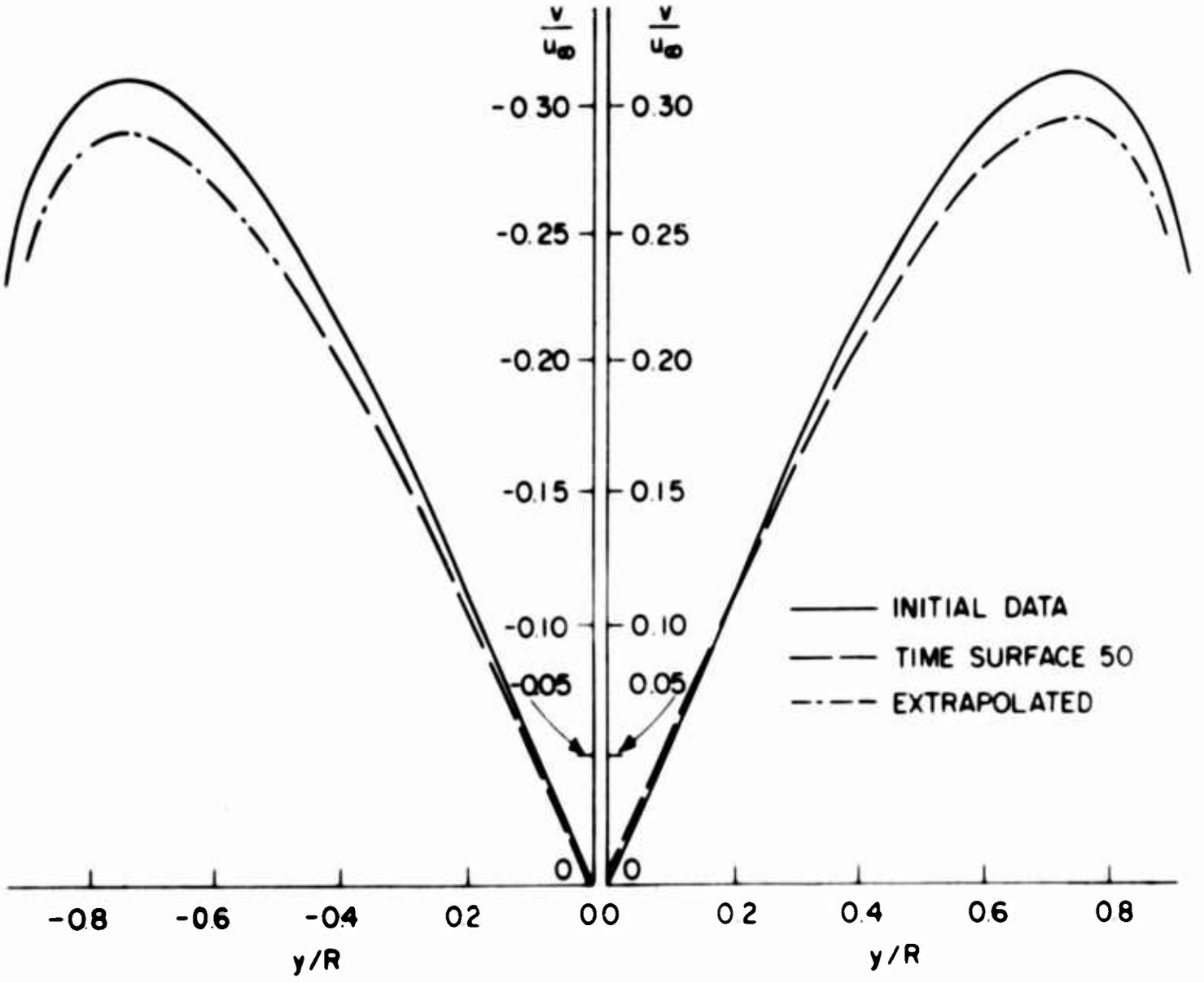


FIGURE 24 BODY SURFACE v VELOCITY COMPONENT FOR STEADY FLOW

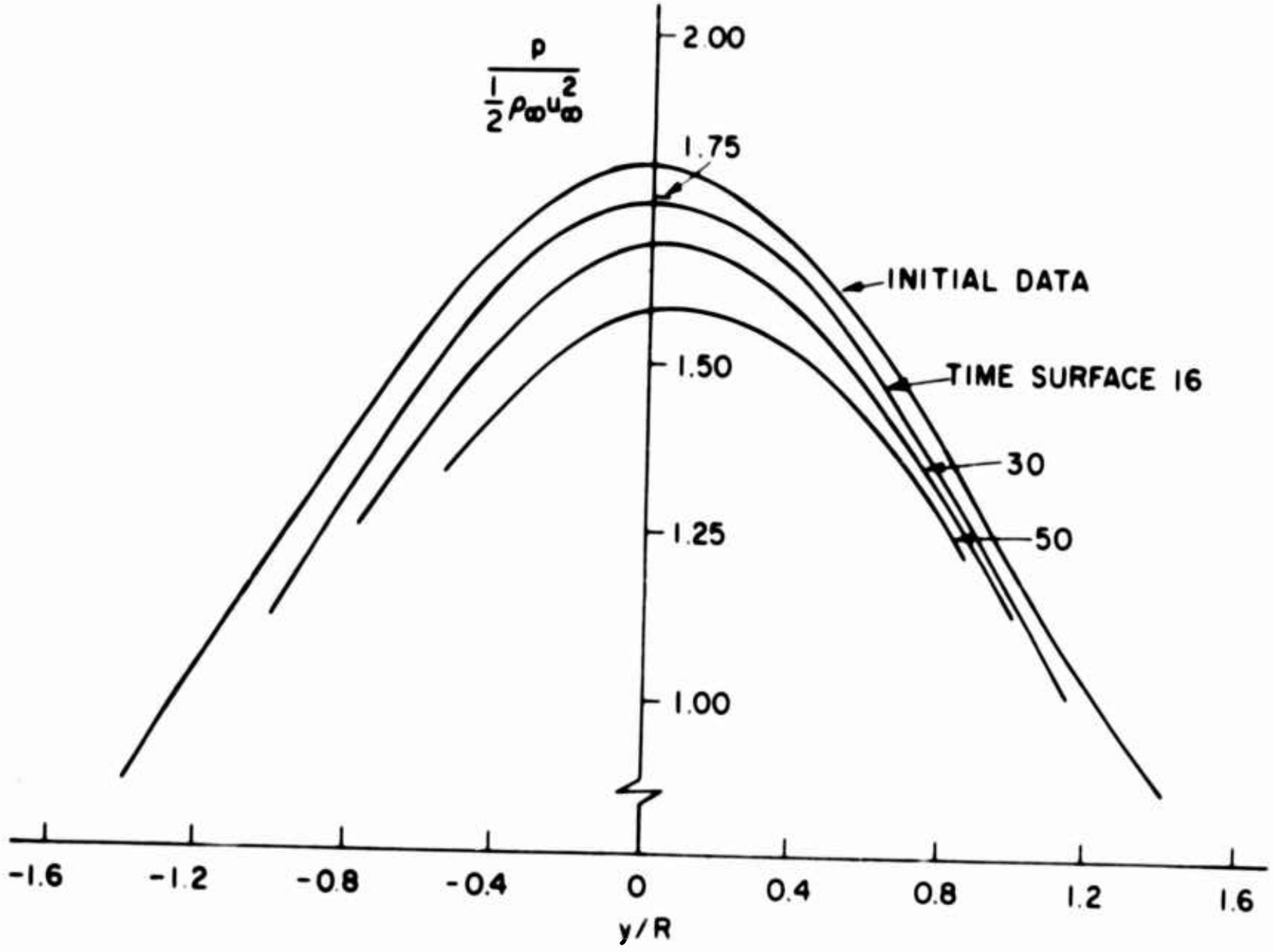


FIGURE 25 PRESSURE ON THE MIDDLE RING OF FIELD POINTS FOR STEADY FLOW

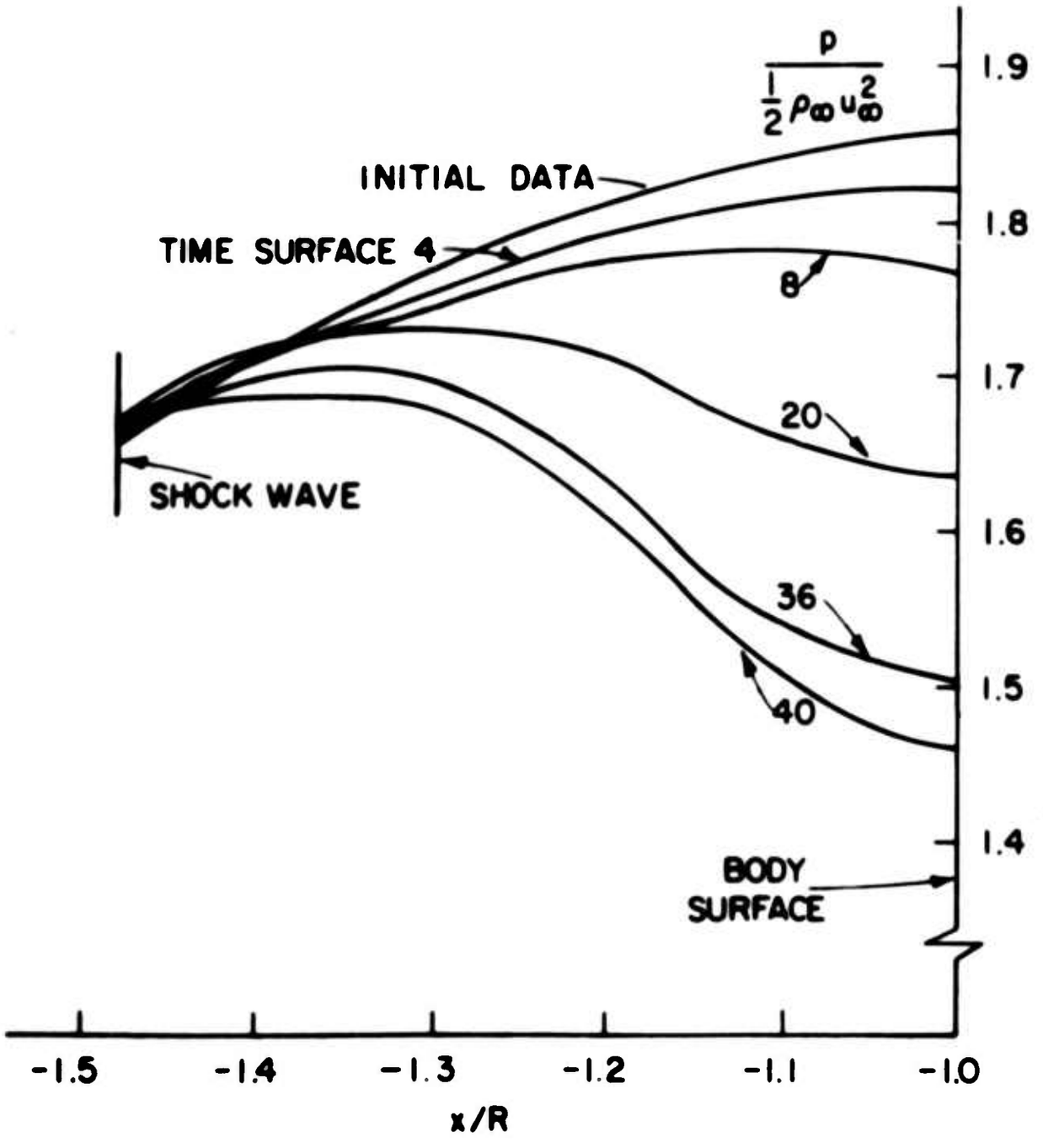


FIGURE 26 PRESSURE ALONG THE AXIS OF SYMMETRY FOR STEADY FLOW

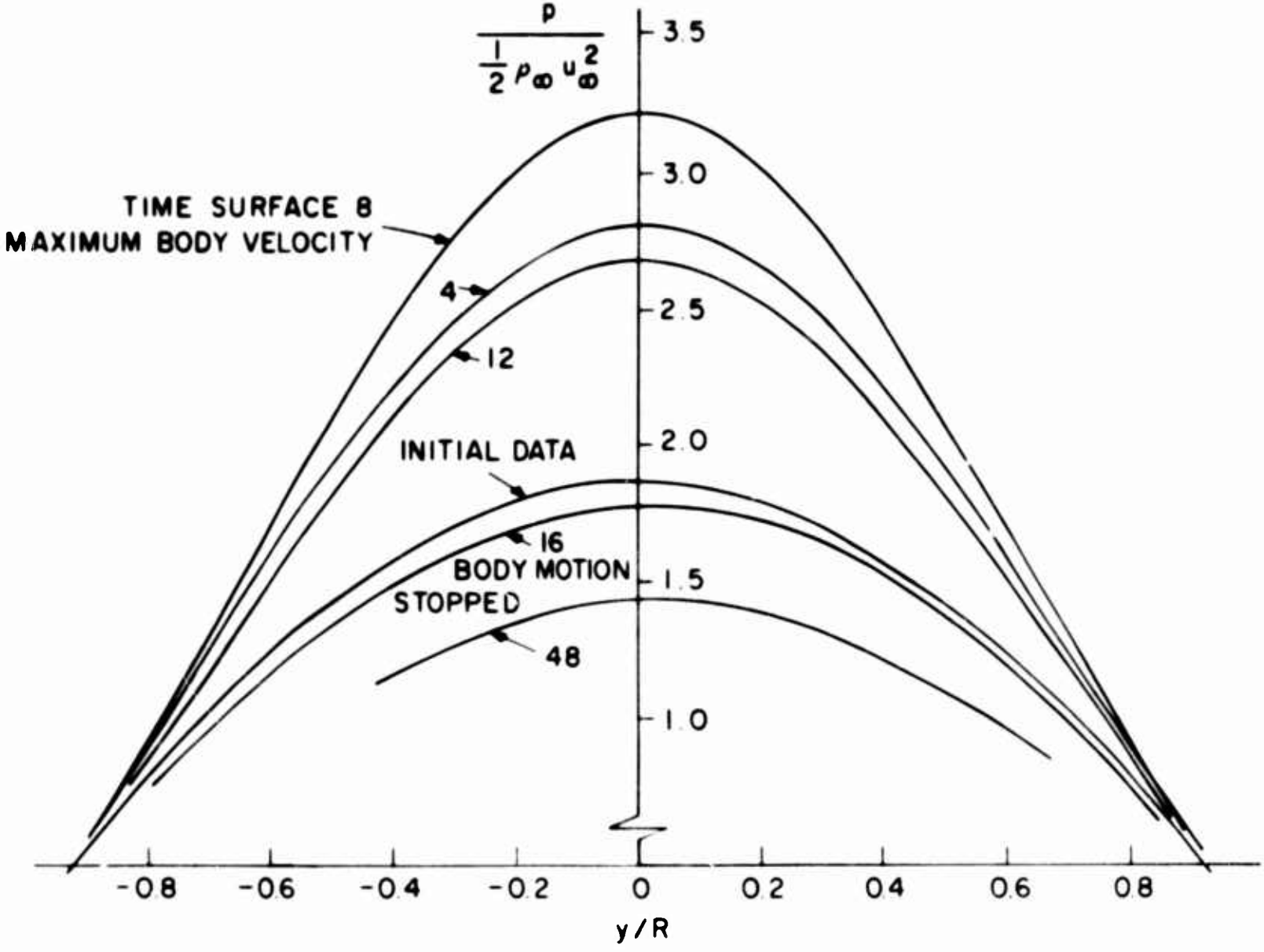


FIGURE 27 BODY SURFACE PRESSURE FOR SYMMETRIC WARPING

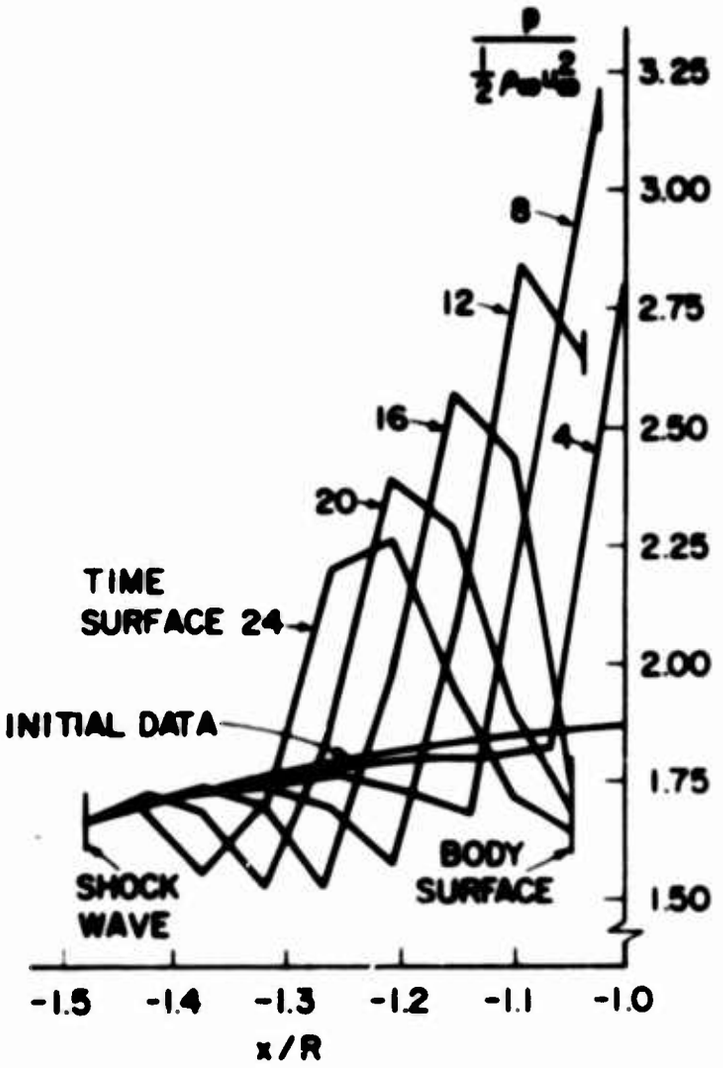


FIGURE 28 (a)

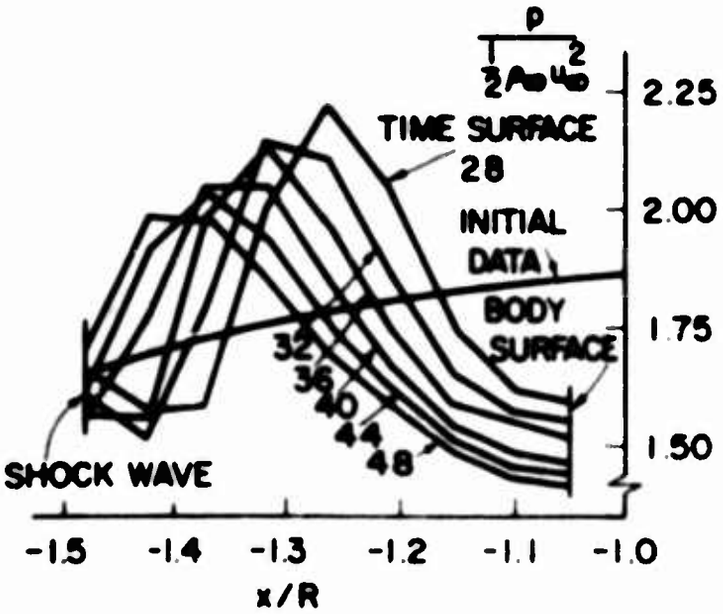


FIGURE 28 (b)

FIGURE 28 PRESSURE ON THE AXIS OF SYMMETRY FOR SYMMETRIC WARPING

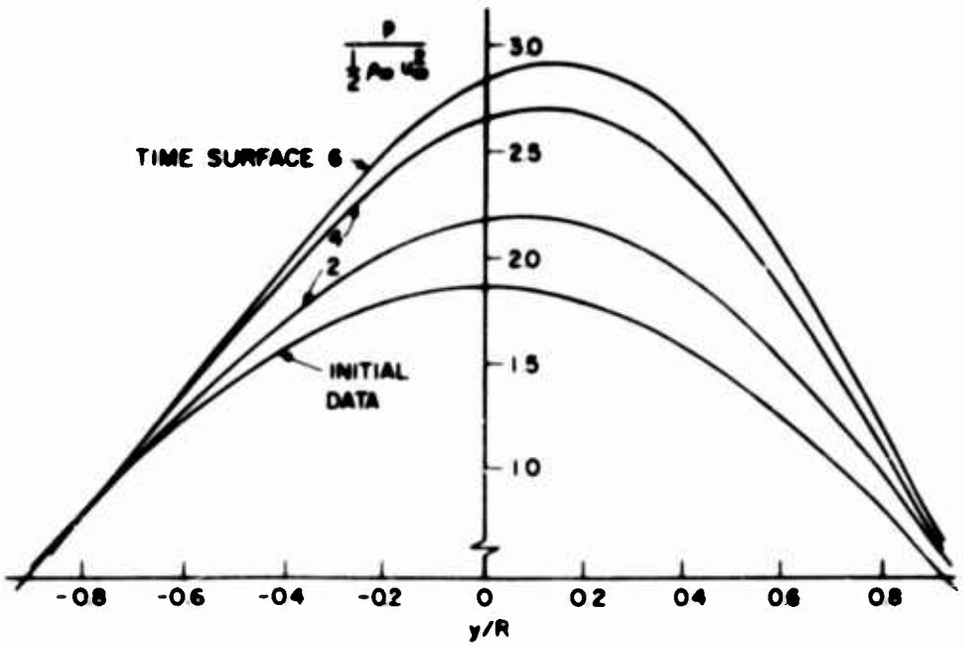


FIGURE 29(a)

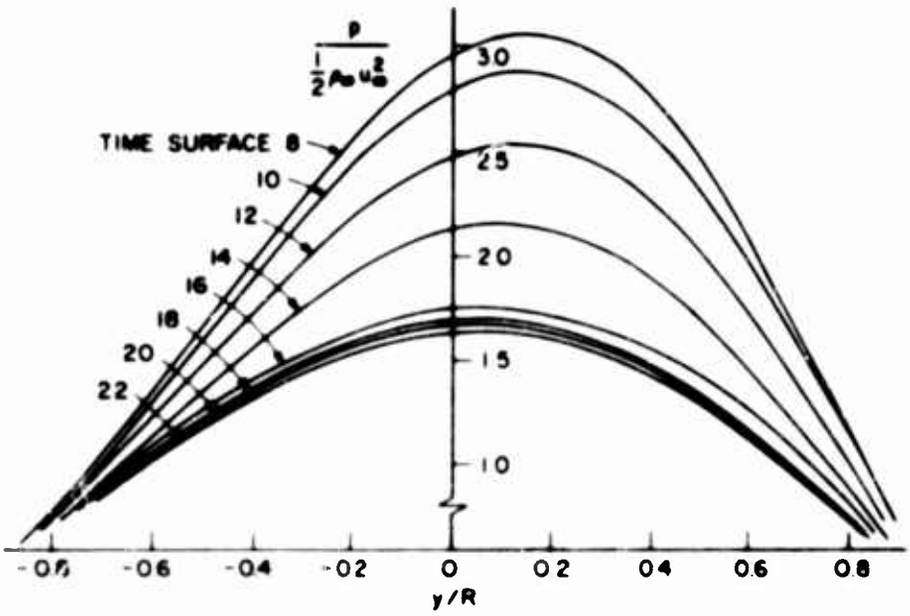


FIGURE 29(b)

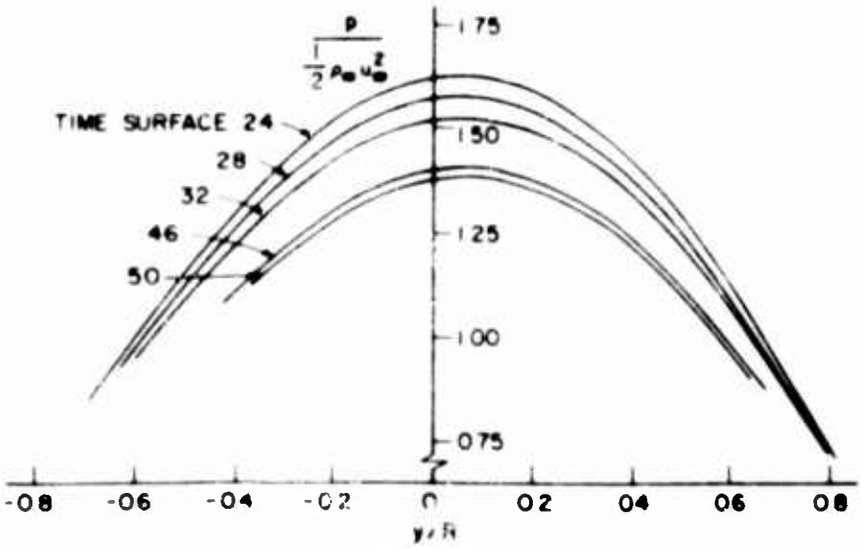
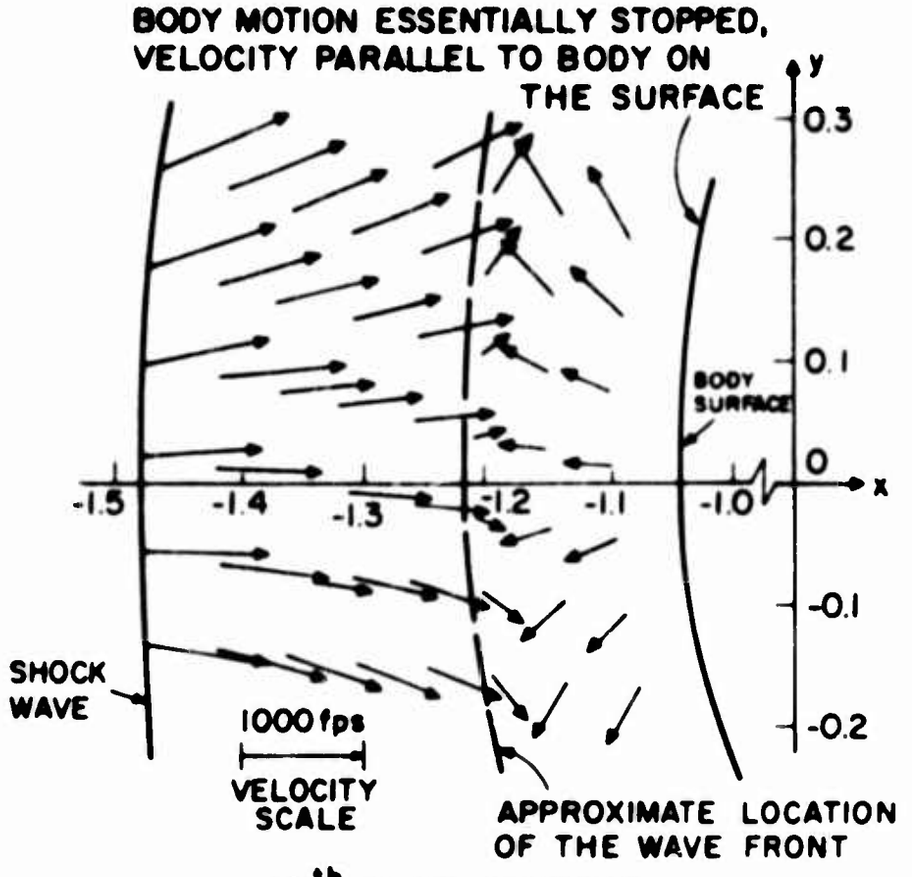


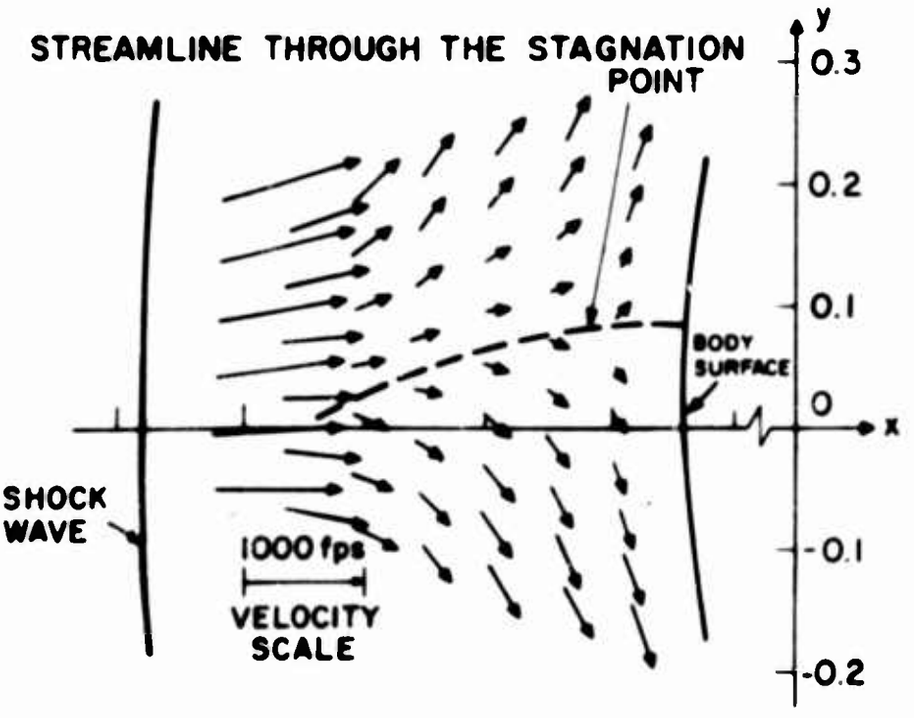
FIGURE 29(c)

FIGURE 29 BODY SURFACE PRESSURE FOR ASYMMETRIC WARPING



16th TIME SURFACE

(a)



34th TIME SURFACE

(b)

FIGURE 30 VELOCITY FIELDS FOR ASYMMETRIC WARPING

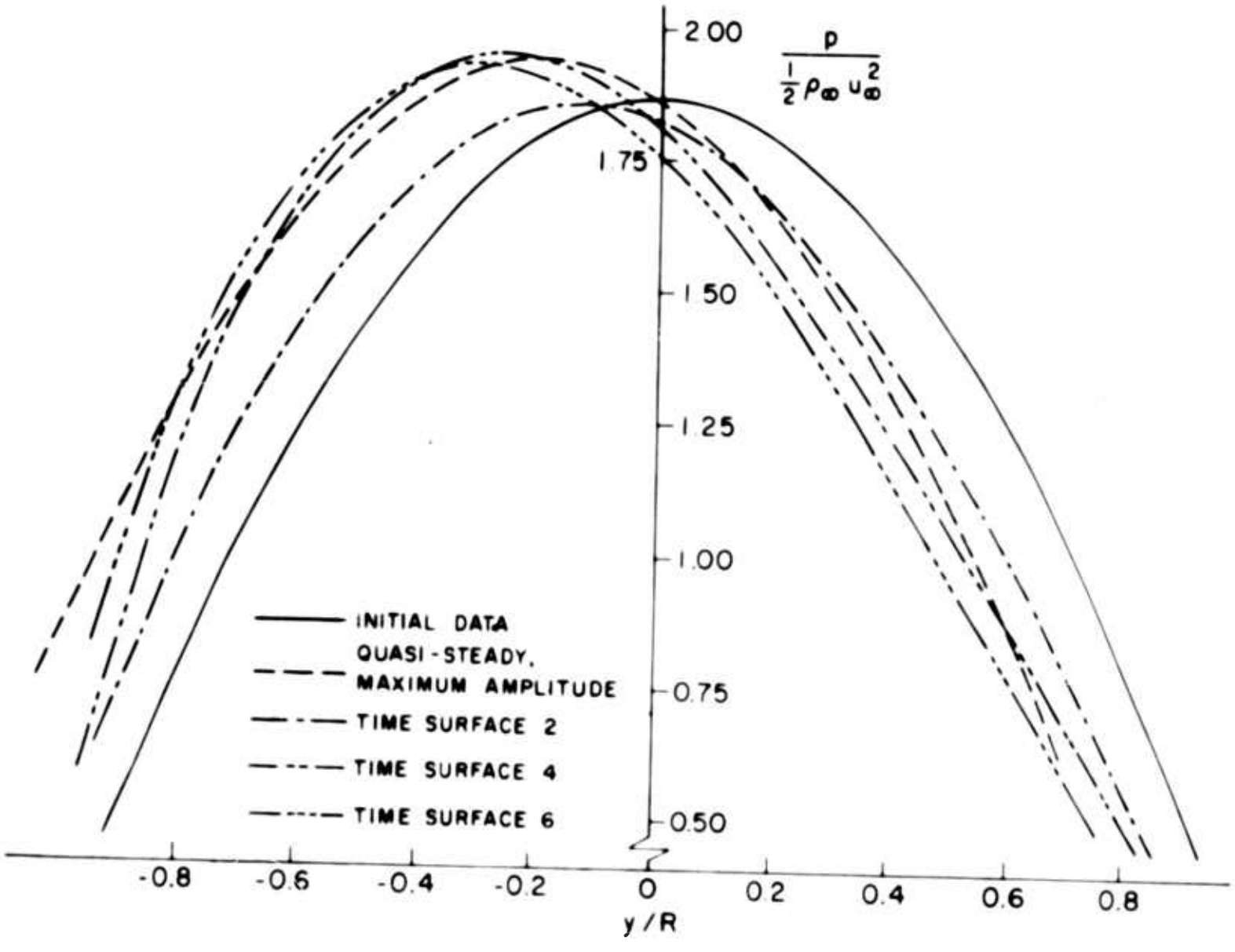


FIGURE 31(a) BODY SURFACE PRESSURE FOR THE VERTICAL OSCILLATION

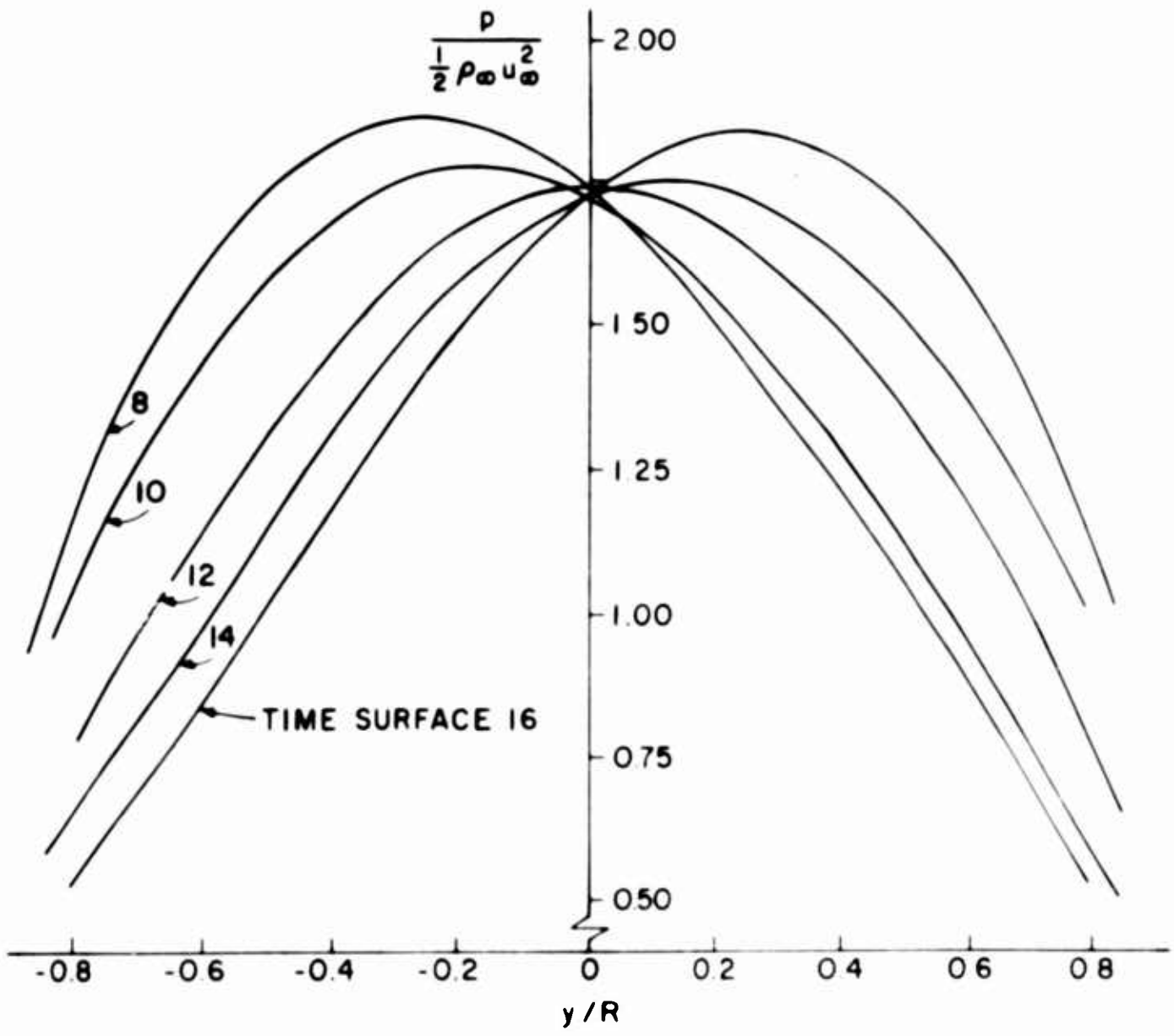


FIGURE 31(b) BODY SURFACE PRESSURE FOR THE VERTICAL OSCILLATION

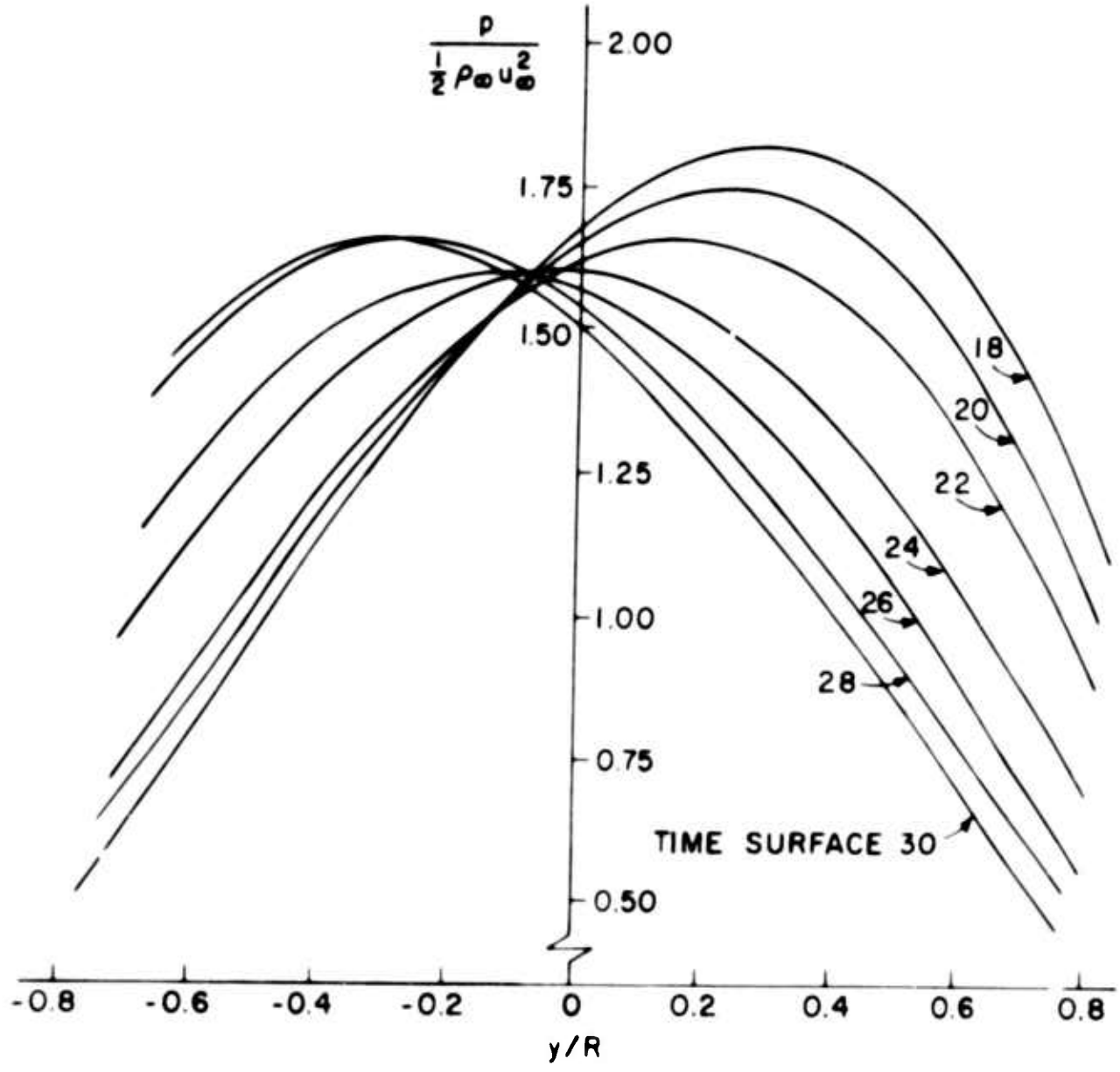


FIGURE 31(c) BODY SURFACE PRESSURE FOR THE VERTICAL OSCILLATION

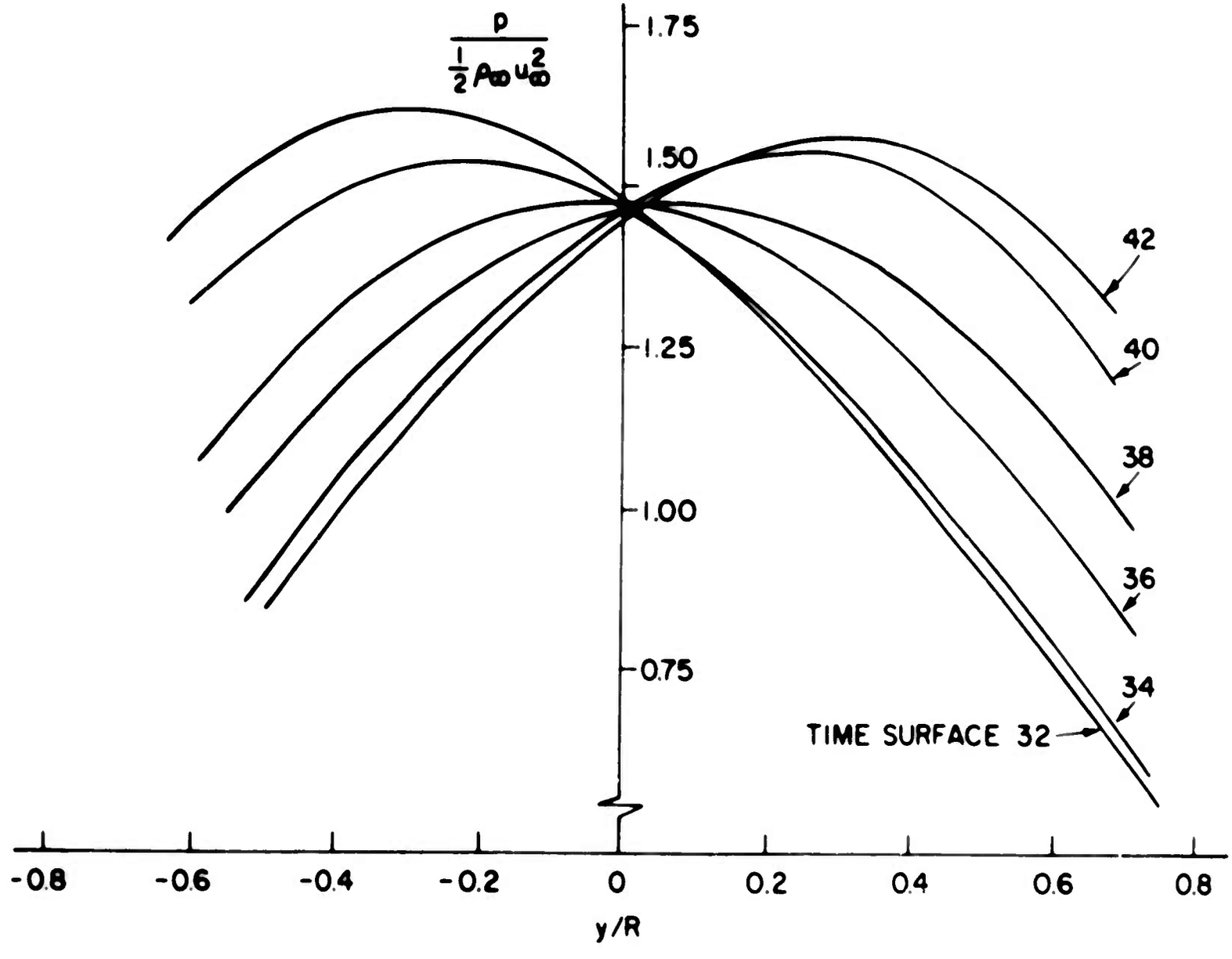


FIGURE 31 (d) BODY SURFACE PRESSURE FOR THE VERTICAL OSCILLATION

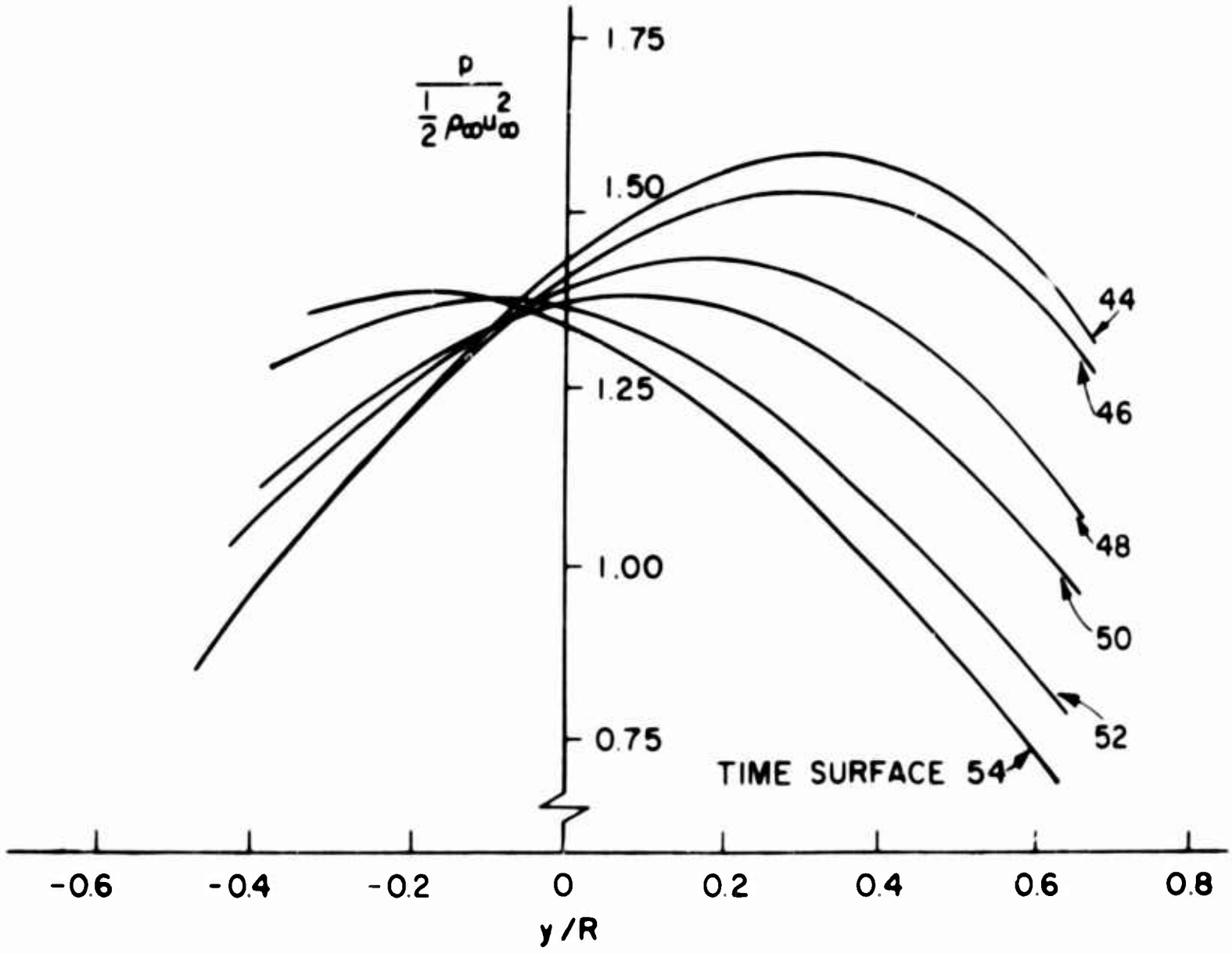


FIGURE 31(e) BODY SURFACE PRESSURE FOR THE VERTICAL OSCILLATION

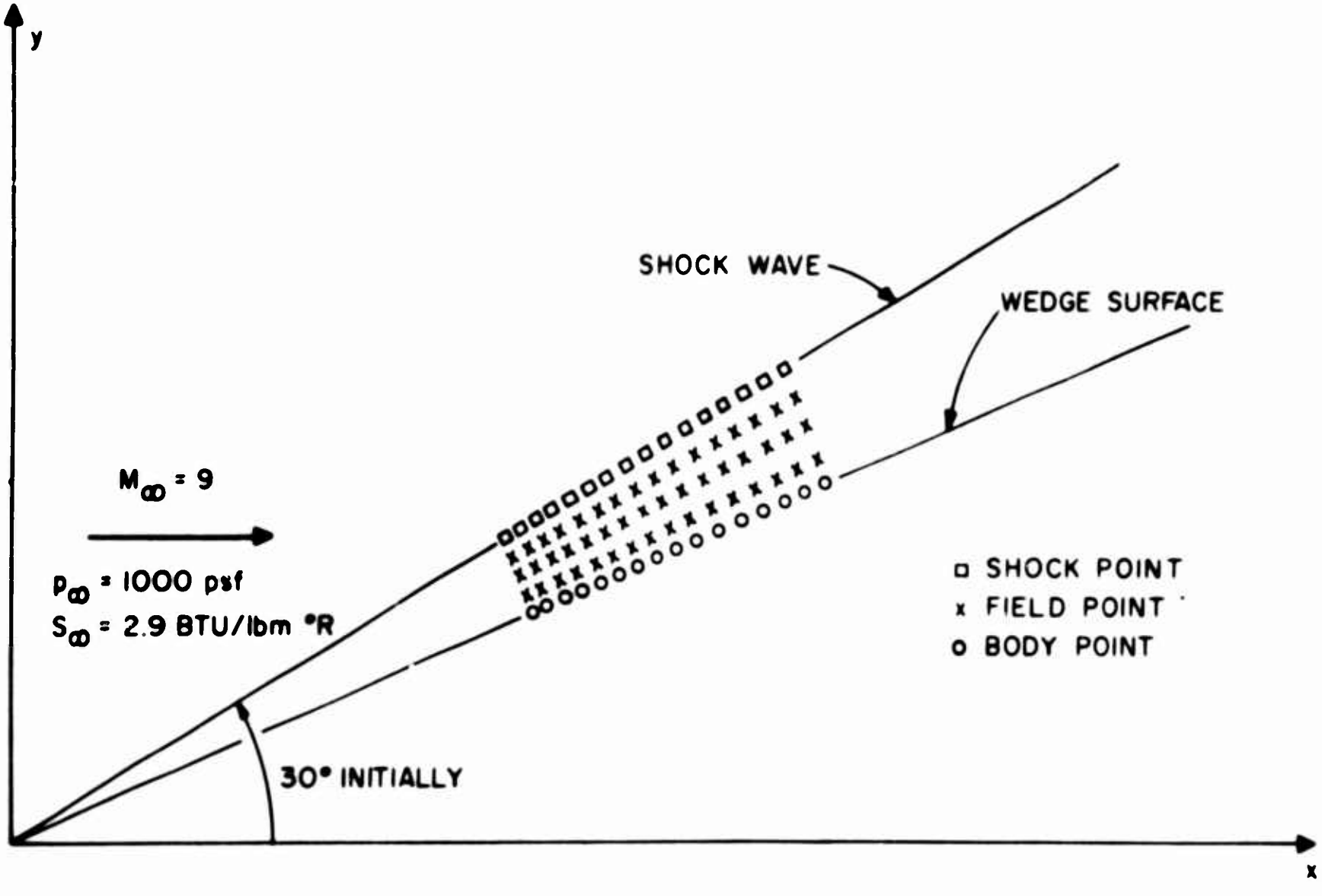
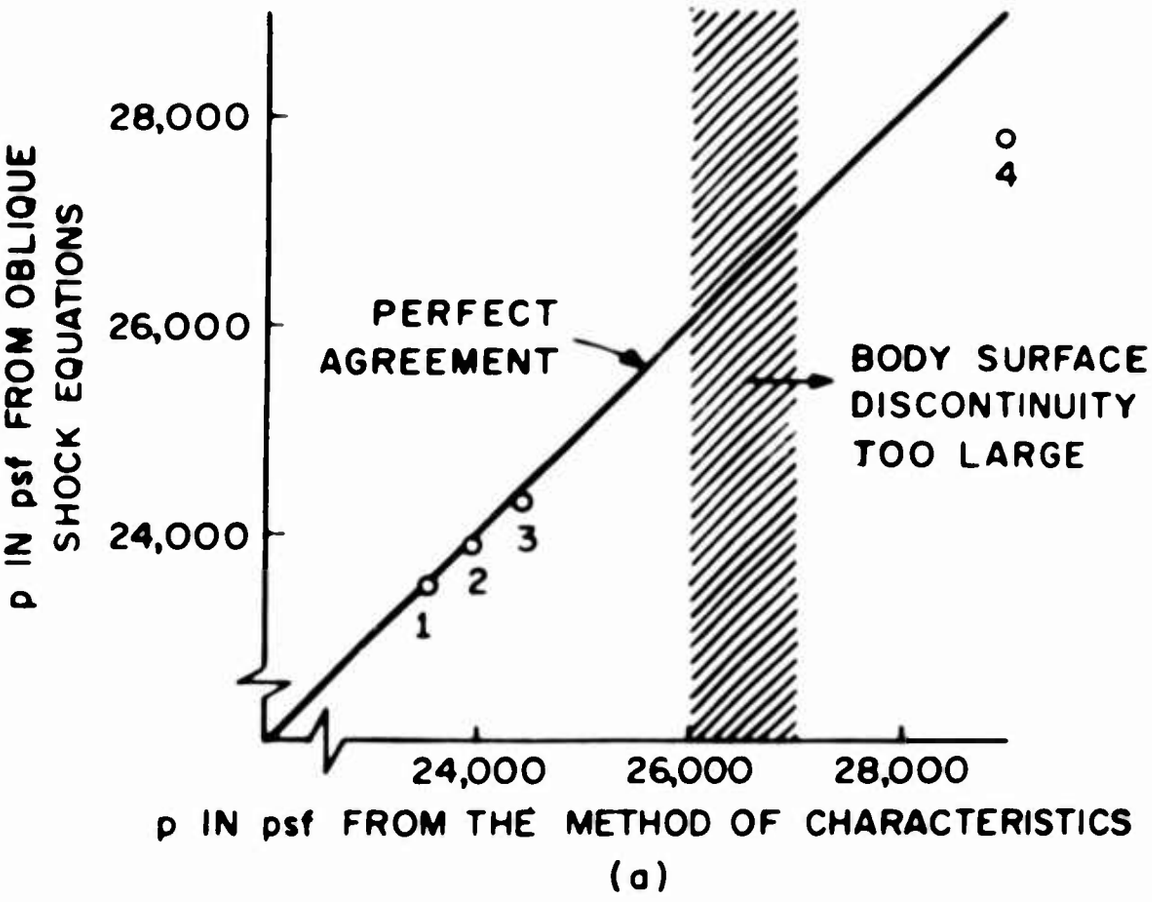
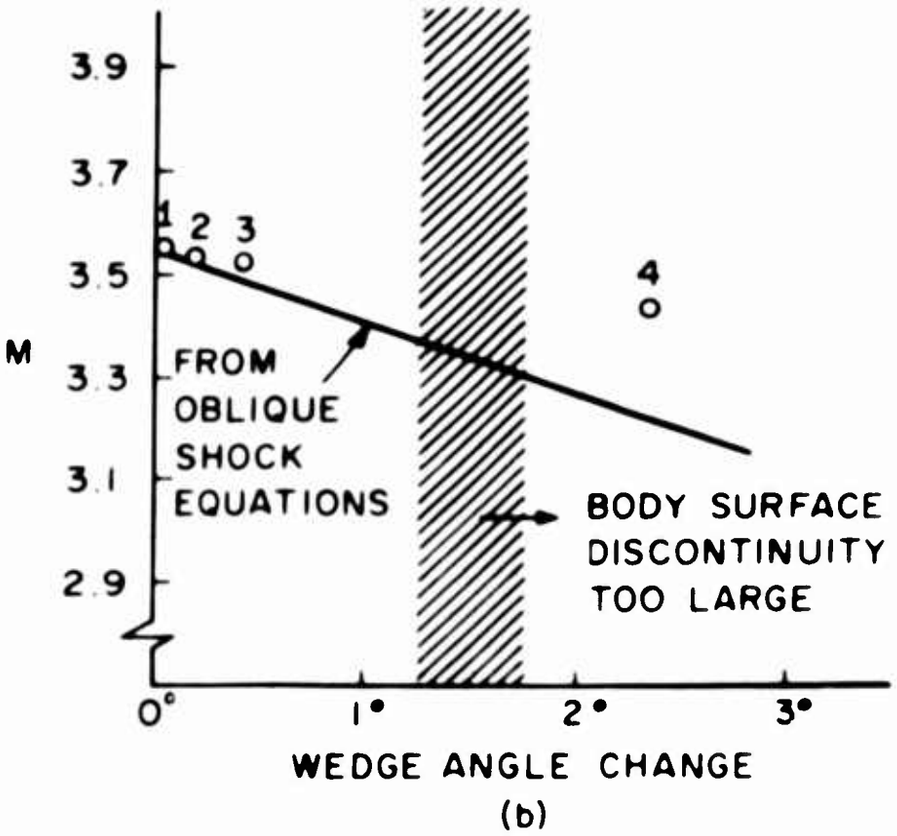


FIGURE 32. TYPICAL INITIAL SURFACE NET CONFIGURATION FOR WEDGE FLOW



(a)



(b)

FIGURE 33 WEDGE FLOW RESULTS

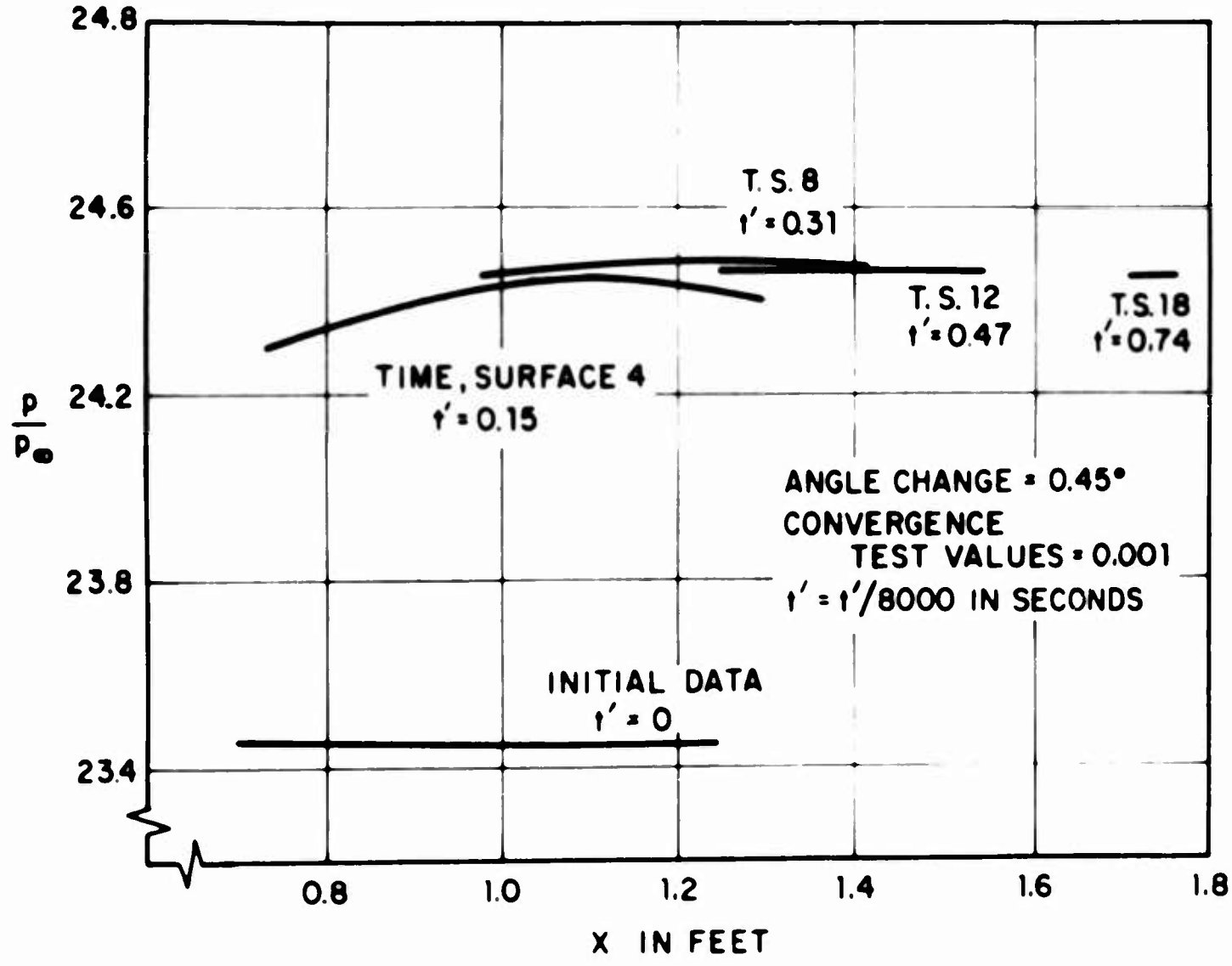


FIGURE 34 TYPICAL PRESSURE DISTRIBUTIONS ON THE WEDGE SURFACE.

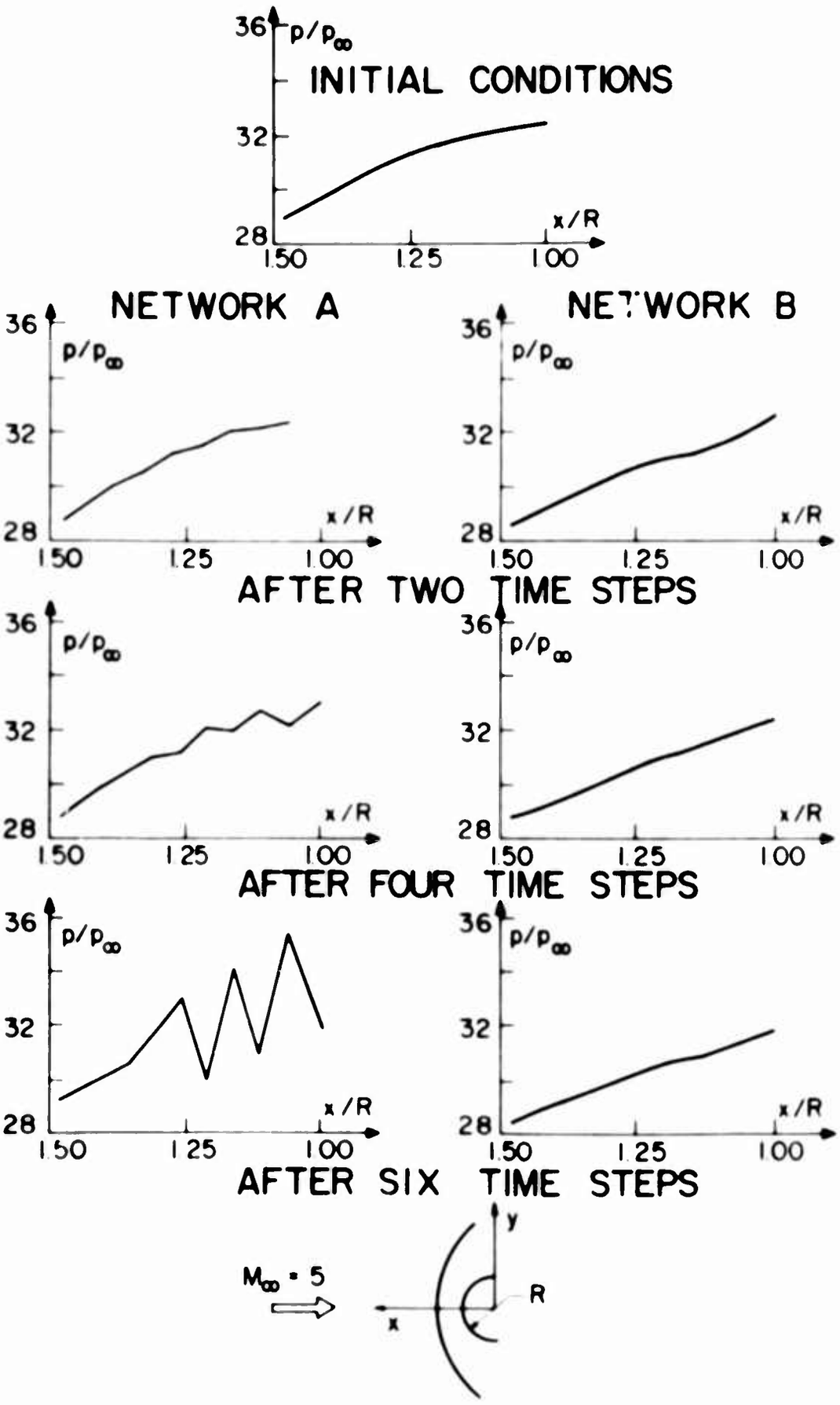


Figure 35 Pressure Near the Axis of Symmetry for the Blunt Body Steady Flow

References

1. Hayes, Wallace D., and Probstein, Ronald F., Hypersonic Flow Theory, Academic Press, New York and London, 1959.

2. Ralston, Anthony, and Wilf, Herbert S., Eds. Mathematical Methods for Digital Computers, John Wiley & Sons, Inc., New York, 1960.

3. von Neumann, J. and Richtmyer, R. D., "A Method for the Numerical Calculation of Hydrodynamic Shocks," Journal of Applied Physics, Vol. 21, 1950, pp. 232-237.

4. Lax, P. D., "Weak Solutions of Nonlinear Hyperbolic Equations and Their Numerical Computation," Commun. on Pure & Applied Math., Vol. VII, 1954, pp. 159-193.

5. Lax, P. D. and Wendroff, B., "Systems of Conservation Laws," Commun. Pure & Applied Math., Vol. 13, 1960, pp. 217-237.

6. Hartree, Douglas R., "Some Practical Methods of Using Characteristics in the Calculation of Non-Steady Compressible Flow," United States Atomic Energy Commission Report AECU-2713, September 1953.

7. Fox, L., Numerical Solution of Ordinary and Partial Differential Equations, Addison-Wesley Publishing Co., Inc., 1962.

8. Forsythe, G. E. and Wasow, W. R., Finite-Difference Methods for Partial Differential Equations, Wiley, New York, 1960.

9. Richtmyer, R. D., Difference Methods for Initial-Value Problems, Interscience Publishers, New York, 1957.

10. Collatz, L., The Numerical Treatment of Differential Equations, Springer-Verlag, Berlin, Göttingen, Heidelberg, 1960.

11. Massau, J., Mémoire sur l'intégration graphique des équations aux dérivees partielles, F. Meyer-Van Loo, Ghent, 1899.

12. Prandtl, L., and Busemann, A., "Naherungsverfahren zur zeichnerischen Bestimmung von ebenen Stromungen und Überschallgeschwindigkeiten," Stodola-Festschrift, Zürich, 1929.

13. Ferri, Antonio, "The Method of Characteristics," Section G in General Theory of High Speed Aerodynamics, Vol. VI of High Speed Aerodynamics and Jet Propulsion, Princeton University Press, 1954, pp. 583-669.

14. Meyer, R. E., "Theory of Characteristics of Inviscid Gas Dynamics," Encyclopedia of Physics, Vol IX, Springer-Verlag, 1960, pp. 225-282.

15. Meyer, R. E., "The Method of Characteristics," in Modern Developments in Fluid Dynamics High Speed Flow, ed. by L. Howarth, Oxford Univ. Press, London, 1953, pp. 71-104.

16. Thornhill, C. K., "The Numerical Method of Characteristics for Hyperbolic Problems in Three Independent Variables," Aeronautical Research Council R. & M. No. 2615, Sept. 1948.

17. Coburn, N. and Dolph, C. L., "The Method of Characteristics in the Three-Dimensional Stationary Supersonic Flow of a Compressible Gas," Proceedings of Symposia in Applied Mathematics, Vol. 1, 1949, pp. 55-56.

18. Sauer, R., "Dreidimensionale Probleme der Charakteristikentheorie Partieller Differentialgleichungen," Zeitschrift Angewandte Mathematik und Mechanik, Bd. 30, Nr. 11/12, Nov. Dez, 1950, p. 347.

19. Clippinger, R. F. and Giese, J. H., "Characteristic Conditions for Three Dimensional Flows with Vorticity," Ballistic Research Laboratories Memorandum Report 615, Aberdeen Proving Ground, Maryland, August 1952.

20. Holt, M., "The Method of Characteristics for Steady Supersonic Rotational Flow in Three Dimensions," JFM, Vol. I, Part 4, Oct. 1956, p. 409.

21. Coburn, N., "Intrinsic Form of the Characteristic Relations for a Perfect Compressible Fluid in General Relativity and Non-steady Newtonian Mechanics," J. Math. Mech. Vol. 9, 1960, pp. 421-437.

22. Saini, G. L., "The Method of Characteristics for Self-Gravitating Anisentropic Relativistic Compressible Fluids," J. Math. Mech., Vol. 10, No. 2, 1961, pp. 223-231.

23. Fowell, Leonard R., "Flow Field Analysis for Lifting Re-entry Configurations by the Method of Characteristics," I.A.S. Paper No. 61-208-1902, June 1961.

24. Sauer, R., "Differenzenverfahren für hyperbolische Anfangswertprobleme bei mehr als zwei unabhangigen Veranderlichen mit Hilfe von Nebencharakteristiken," Numerische Mathematik, Vol. 5, March 1963, pp. 55-67.

25. Moeckel, W. E., "Use of Characteristic Surfaces for Unsymmetrical Supersonic Flow Problems,"NACA TN. 1849, 1949.

26. Ferrari, C., "Interference Between Wing and Body at Supersonic Speeds-Analysis by the Method of Characteristics," JAS, Vol. 16, 1949, pp. 411-434.

27. Bruhn, G. and Haack, W., "Ein Charakteristikenverfahren für dreidimensionale instationare Gasstromungen," Z. angew. Math. Phys., Vol. 9, 1958, pp. 173-190.

28. Butler, D. S., "The Numerical Solution of Hyperbolic Systems of Partial Differential Equations in Three Independent Variables," Proceedings of the Royal Society, Series A, Vol. 255, 1960, pp. 232-252.

29. Tsung, Cheng-Chih, "Study of Three-Dimensional Supersonic Flow Problems by a Numerical Method Based on the Method of Characteristics," Ph.D. Thesis in Mech. Eng., University of Illinois, 1960.

30. Morretti, G., Sanlorenzo. E. A., Magnus, E. E., and Weilerstein, G., "Supersonic Flow about General Three-Dimensional Blunt Bodies," Vol. III, Flow Field Analysis of Reentry Configurations by General Three-Dimensional Method of Characteristics, ASD-TR-61-727, 1962.

31. Chu, Boa-Teh, "Wave Propagation and the Method of Characteristics in Reacting Gas Mixtures with Applications to Hypersonic Flow," WADC TN-57-213, 1957.

32. Broer, L. J. F., "Characteristics of the Equations of Motion of a Reacting Gas," JFM, Vol. 4, 1958, pp. 276-282.

33. Sedney, R., South, J. C., and Gerber, N., "Characteristic Calculation of Non-Equilibrium Flows," Proceedings of the AGARD Meeting on High Temperature Effects on Hypersonic Flow, Pergamon Press, April 1962.

34. Sedney, R., and Gerber, N., "Nonequilibrium Flow Over a Cone," IAS Paper No. 63-71, 1963.

35. Capiaux, R. and Washington, M., "Nonequilibrium Flow Past a Wedge," AIAA Journal, Vol. 1, 1963, pp. 650-661.

36. Wood, A. D., Springfield, J. F., and Pallone, A. J., "Determination of the Effects of Chemical and Vibrational Relaxation on an Inviscid Hypersonic Flow Field," AIAA Paper No. 63-441 presented at the Conference on Physics of Entry into Planetary Atmospheres, Cambridge, Mass., 1963.

37. Courant, R. and Hilbert, D., Methods of Mathematical Physics, Vol. II, Partial Differential Equations, Interscience Publishers, New York & London, 1962.

38. von Mises, Richard, Mathematical Theory of Compressible Fluid Flow, Vol. 3 in the series on Applied Mathematics and Mechanics, Academic Press, New York, 1958.

39. Li, Ting Y., "Recent Advances in Nonequilibrium Dissociating Gasdynamics," ARSJ, Vol. 31, 1961, pp. 170-178.

40. Holt, M., "The Method of Near Characteristics for Unsteady Flow Problems in Two Space Variables," Institute of Engineering Research, University of California, Report No. AS-63-2, June 1963.

41. Titt, E. W., "An Initial Value Problem for all Hyperbolic Partial Differential Equations of Second Order with Three Independent Variables," Ann. Math (2), Vol. 40. 1939, p. 862.

42. Lapidus, Leon, Digital Computation for Chemical Engineers, McGraw-Hill Book Co., Inc., New York, 1962.

43. Ferri, A., Libby, P. A., and Zakkay, V., "Theoretical and Experimental Investigation of Supersonic Combustion," Aeronautical Research Laboratories, U.S.A.F., Report No. ARL 62-467, Sept. 1962.

44. Moeckel, W. E., "Oblique-Shock Relations at Hypersonic Speeds for Air in Chemical Equilibrium," NACA TN 3895, Jan. 1957.

45. Talbot, G. P., "Application of the Numerical Method of Characteristics in Three Independent Variables to Shock-Thermal Layer Interaction Problems," A.R.D.E. Report (B) /63.

46. General Information Manual, FORTRAN, International Business Machines Corp., 1961.

47. Reference Manual, IBM 7090/7094 Data Processing System, International Business Machines Corp., 1963.

48. Van Dyke, M. D., "A Study of Hypersonic Small-Disturbance Theory," NACA Report 1194, 1954.

49. Belotserkovskii, O. M., "Flow Past a Circular Cylinder with a Detached Shock Wave," Vychislitelnaia Mathematika, Vol. 3, pp. 149-185, 1958; also translated as Avco Corp. Tech. Memo., RAD-9-TM-59-66, September 1959.

50. Ames Research Staff, "Equations, Tables and Charts for Compressible Flow," NACA Report 1135, 1953.

51. Springfield, J. F., private communication.

52. Courant, R., Friedrichs, K. O., and Lewy, H., "Über die partiellen Differenzengleichungen der mathematischen Physik," Math. Ann., Vol. 100, 1928, pp. 32-74.

53. Hahn, S. G., "Stability Criteria For Difference Schemes," Commun. on Pure & Applied Math., Vol. XI, 1958, pp. 243-255.

54. Lax, P. D., "Differential Equations, Difference Equations and Matrix Theory," Commun. on Pure & Applied Math., Vol. XI, 1958, pp. 175-194.

55. Strang, G., "Accurate Partial Difference Methods II: Non-linear Problems," Numerische Mathematik, Vol. 6, 1964.